

A I A Academy  
\_JavaProject

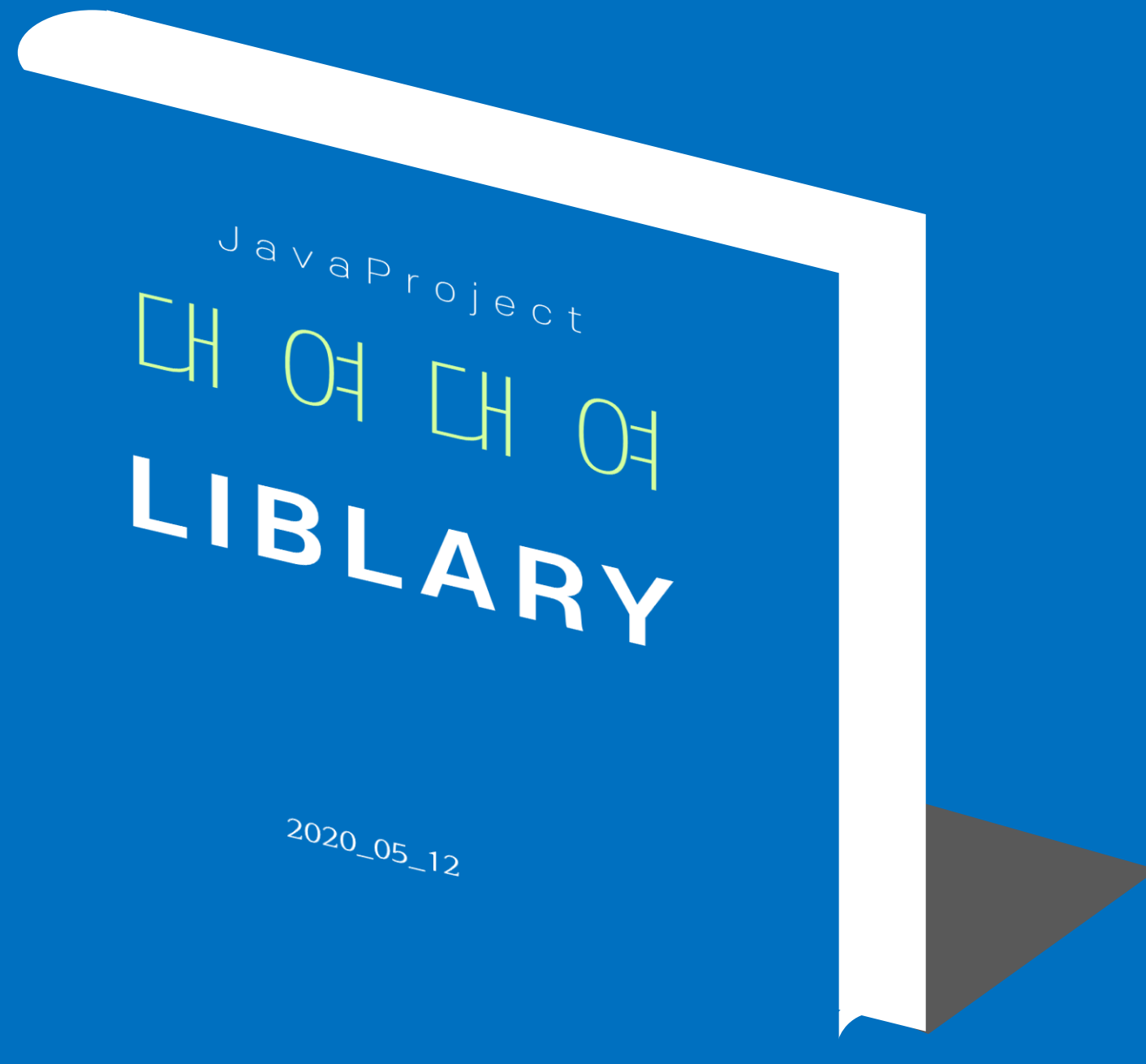
프로젝트 주제  
\_대여관리 프로그램

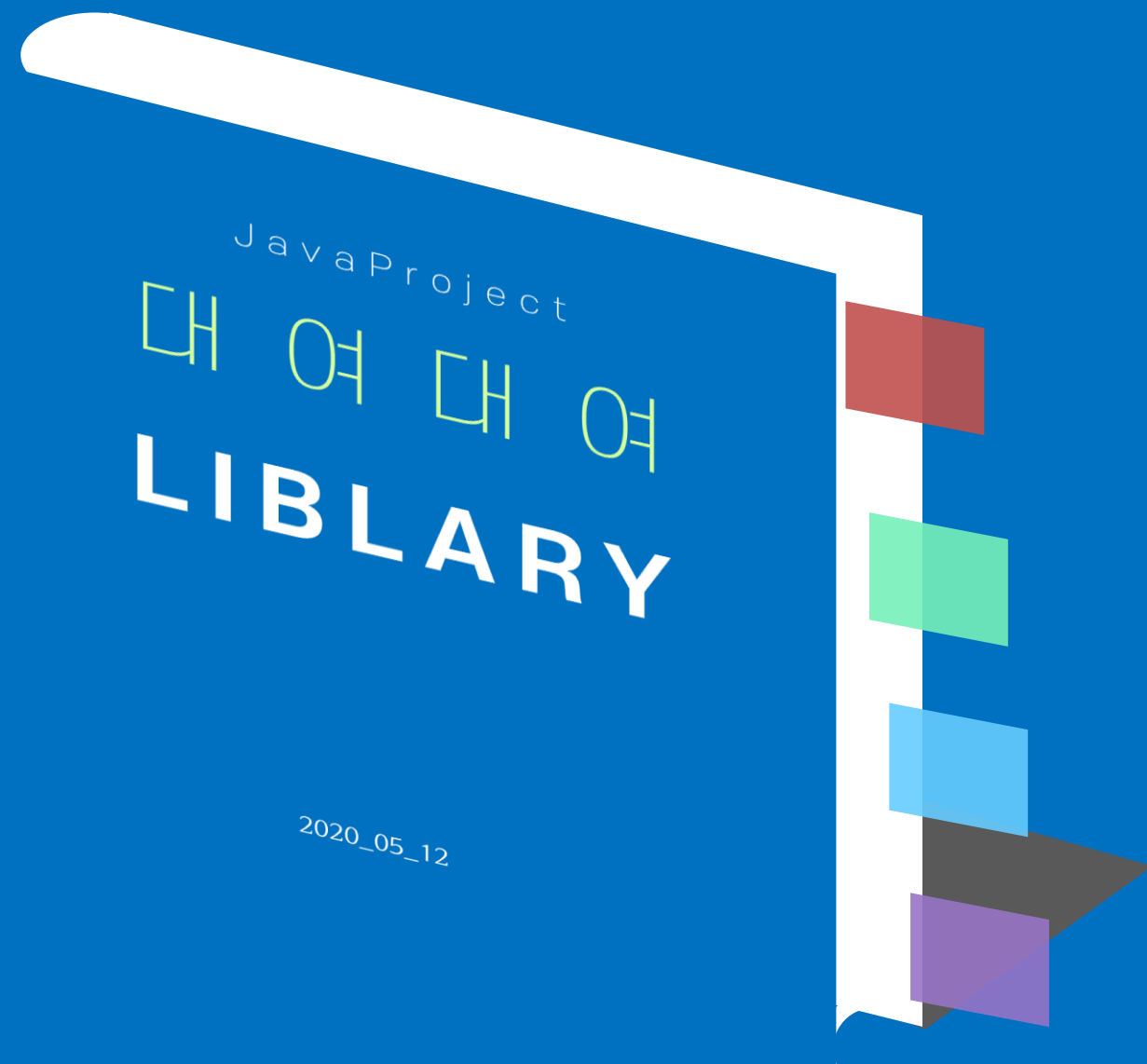
발표일  
\_2020. 05. 12

팀명  
\_대여대여

팀장  
\_임재영

팀원  
\_박명훈  
\_박혜미  
\_박성민  
\_임재영 96





01\_프로그램 흐름도 / 클래스 설계 / 시연 p3

02\_기능별 소개 p7

03\_협업과정에서의 문제점 & 해결방법 p46

04\_프로젝트 개선 방향 p48



## ◆ 프로그램 흐름도 / 클래스 설계 / 시연

### 케이스 1. 관리자시점

관리자로그인 — 회원관리 . 자료관리 . 대여내역관리 — 추가수정삭제

### 케이스 2. 비회원시점

비회원 — 이용안내회원가입 — 검색대여 — 대여

### 케이스 3. 회원

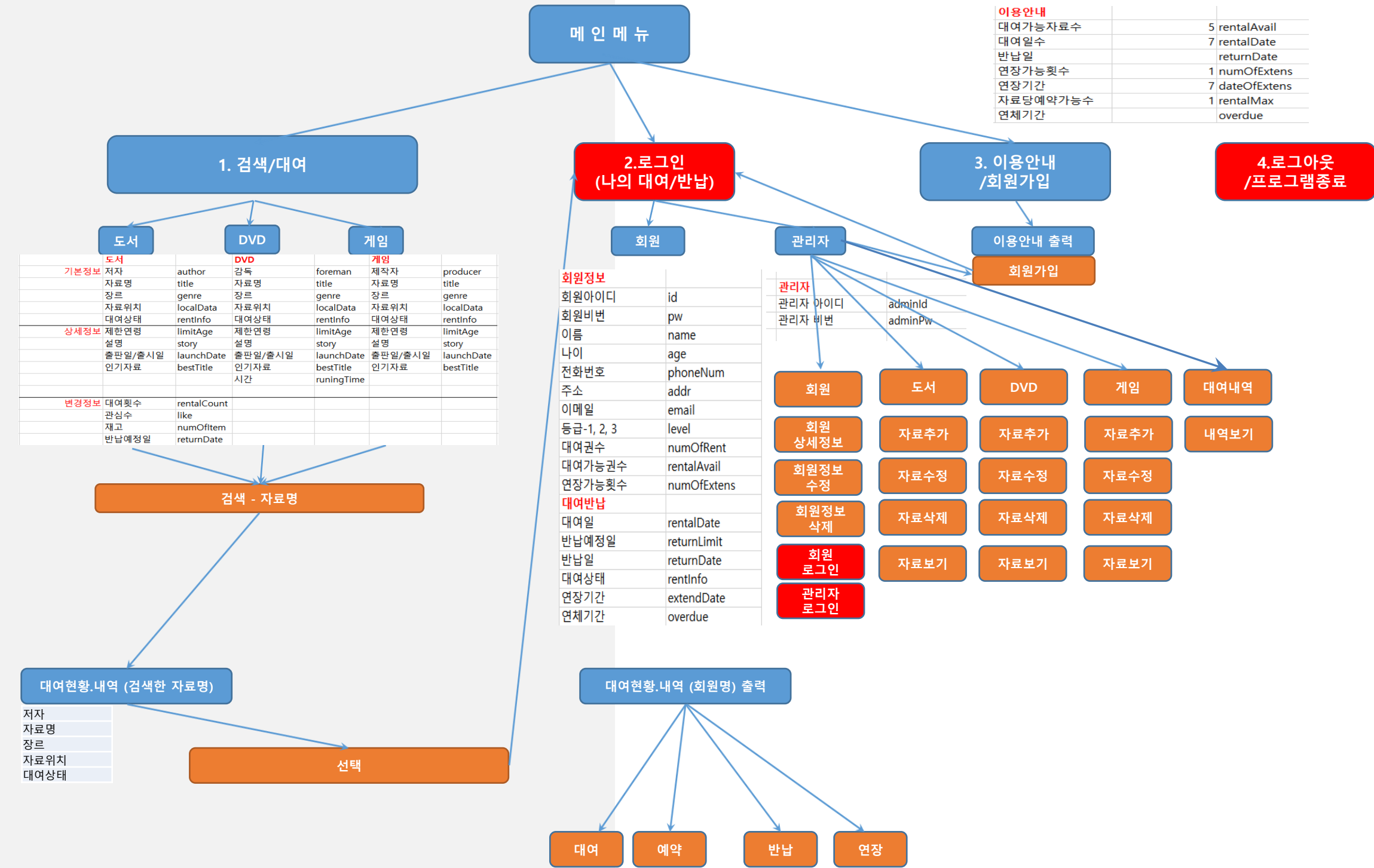
검색대여 — 대여 . 예약 — 내대여내역확인 — 반납연장

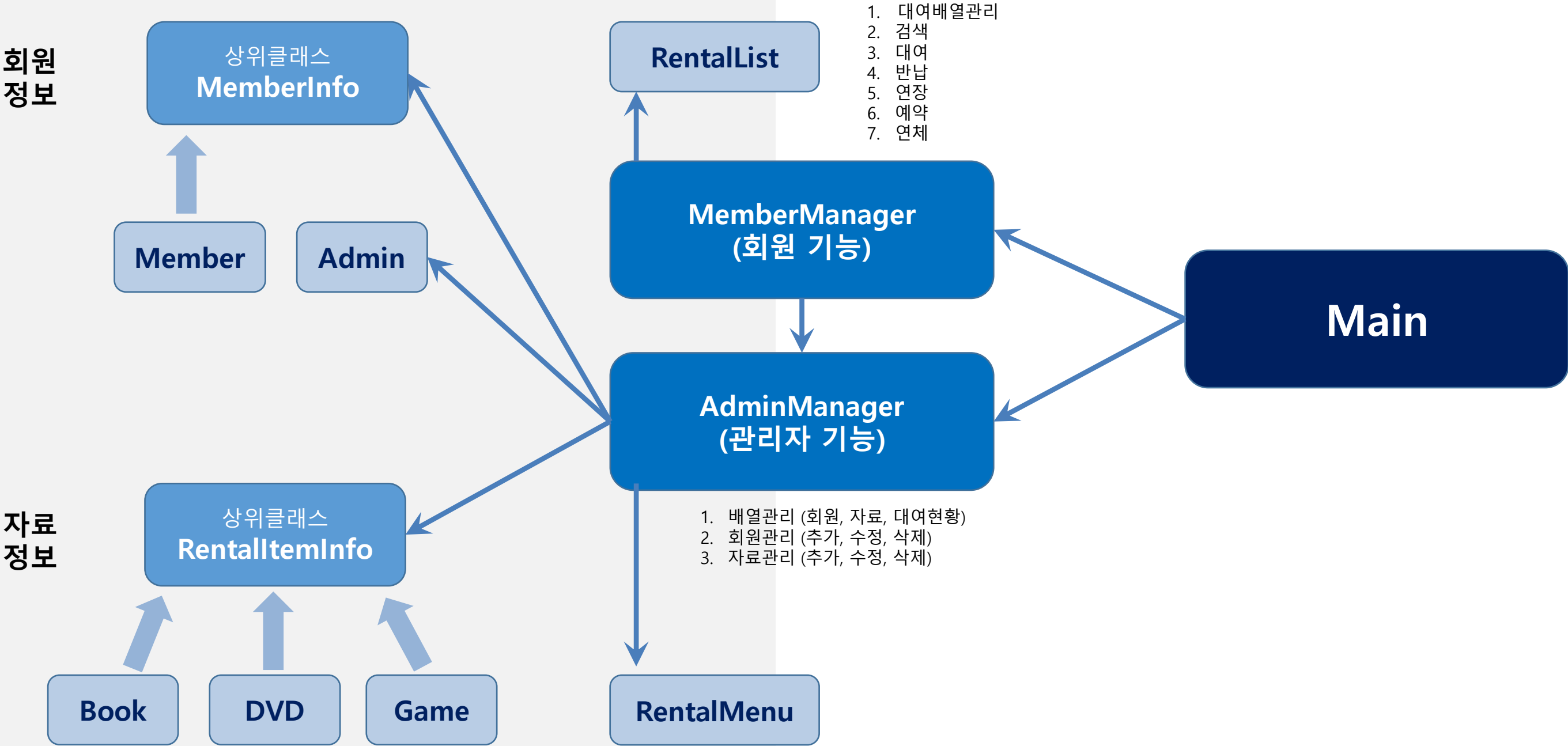
# 프로그램 흐름 / 클래스 설계 / 시연

## : 프로그램 흐름도

### 범례

- 메뉴구성
- 필요기능
- 필요 변수명



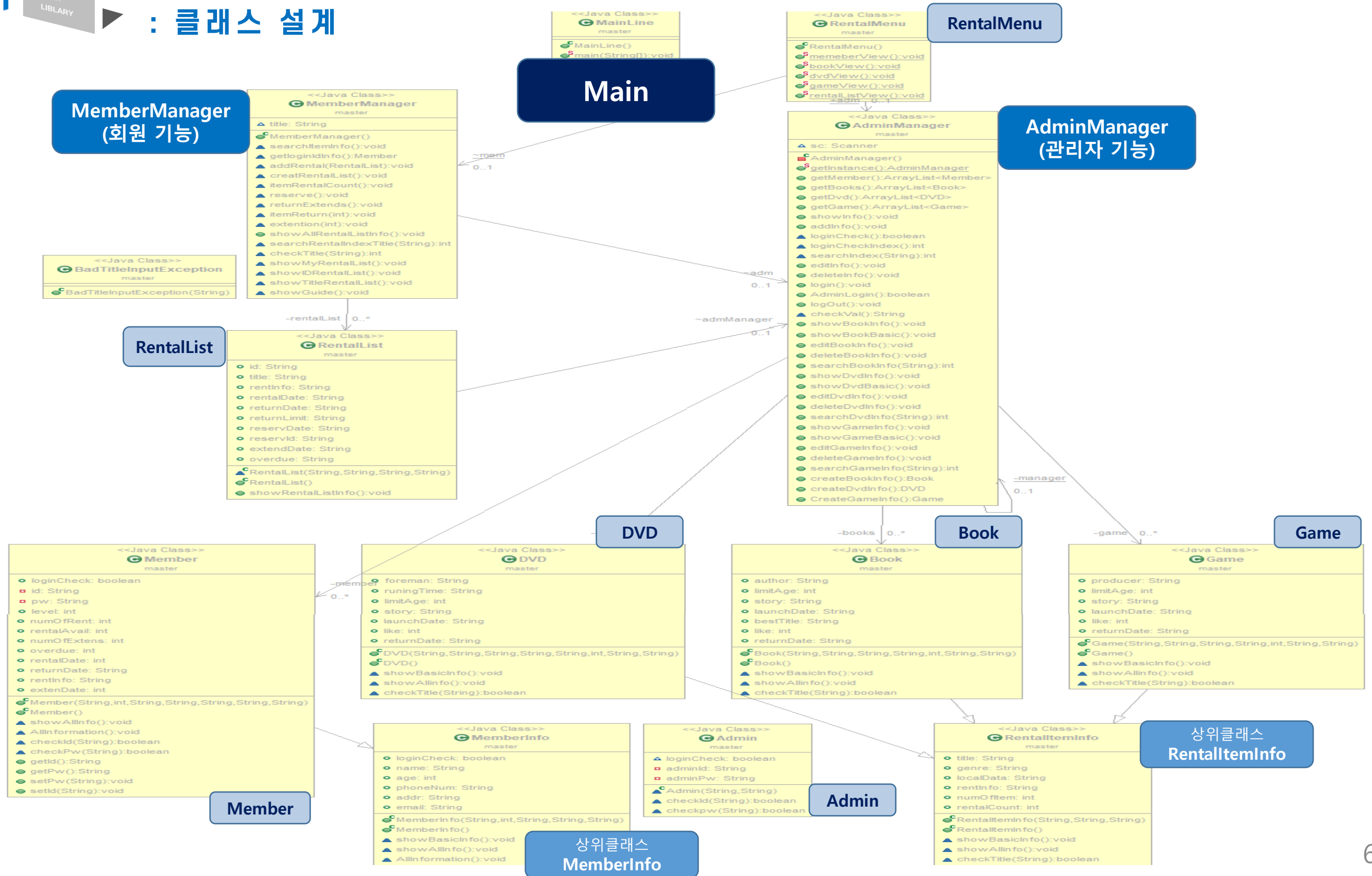


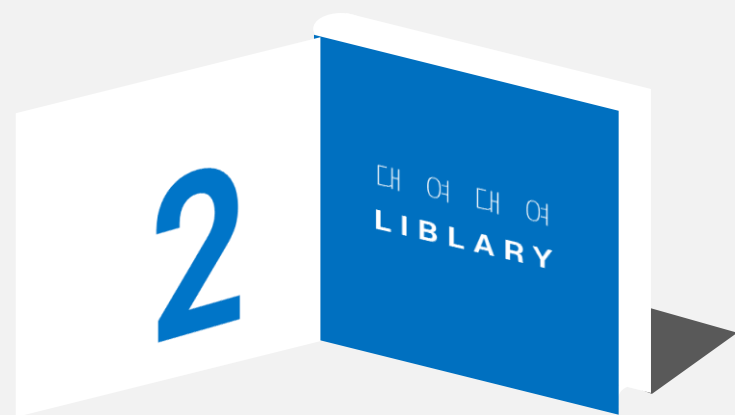
01

대  
여  
대  
여  
LIBRARY

## 프로그램 흐름 / 클래스 설계 / 시연

## : 클래스 설계





## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

\_박명훈

회원, 자료, 대여내역 정보추가

### 3) 관리자매니저

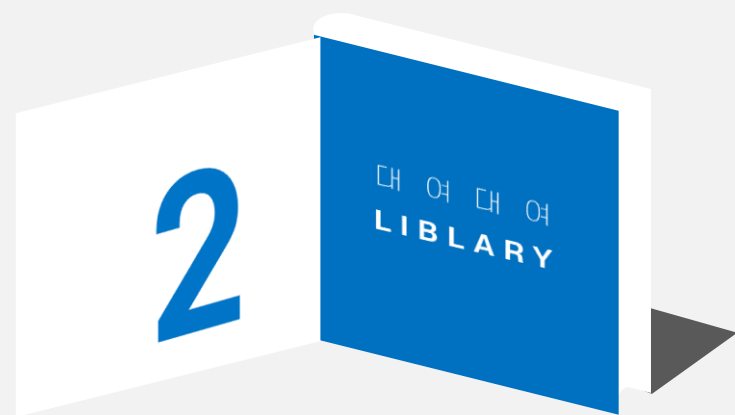
\_박성민, 임재영<sup>96</sup>

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저

\_박혜미, 임재영

검색 / 대여 / 예약 / 반납 / 연장 / 연체



## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

회원, 자료, 대여내역 정보추가

### 3) 관리자매니저

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저

검색 / 대여 / 예약 / 반납 / 연장 / 연체

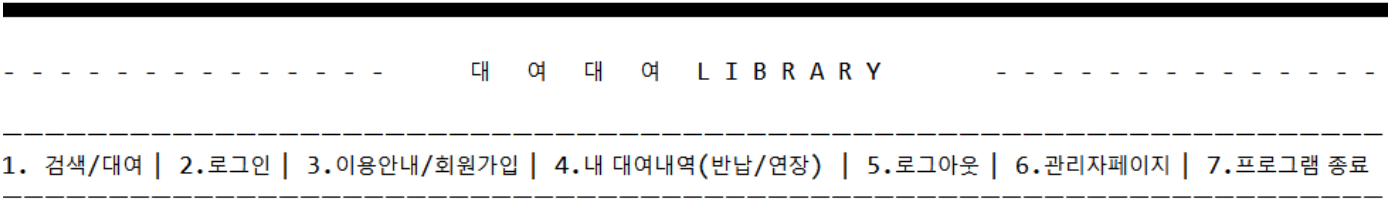




# 기능별 소개

## 1) 메인 \_검색/대여/예약/출력

```
public class MainLine {  
  
    public static void main(String[] args) {  
  
        AdminManager adm = AdminManager.getInstance();  
        MemberManager mem = new MemberManager();  
  
        while (true) {  
            System.out.println();  
            System.out.println("-----");  
            System.out.println();  
            System.out.println("대여대여 LIBRARY");  
            System.out.println();  
            System.out.println("-----");  
            System.out.println("1. 검색/대여 | 2.로그인 | 3.이용안내/회원가입 | 4.내 대여내역(반납/연장) | 5.로그아웃 | 6.관리자페이지 | 7.프로그램 종료");  
            System.out.println("-----");  
            System.out.println();  
  
            int selectNum;  
  
            try {  
                selectNum = adm.sc.nextInt();  
                adm.sc.nextLine();  
            } catch (Exception e) {  
                System.out.println("잘못된 입력입니다. \n1~7사이 숫자를 입력해 주세요.");  
                adm.sc.nextLine();  
                continue;  
            }  
        }  
    }  
}
```



- 1. 검색/대여 : 도서, 게임, DVD를 검색만 가능
- 2. 로그인 : 회원가입을 해야지만 도서, 게임, DVD 대여가능
- 3. 이용안내/회원가입 : 이용안내 문구와 회원가입에 필요한 상세정보 입력
- 4. 내 대여내역(반납/연장) : 회원의 대여내역을 확인 또는 반납, 연장 가능
- 5. 로그아웃 : 나가기
- 6. 관리자 페이지 : 관리자로 접속을 하면 개인회원 추가, 수정, 삭제 가능하고 대여 또한 도서, 게임, DVD 추가, 수정, 삭제 가능
- 7. 프로그램 종료



## 기능별 소개

### 1) 메인 \_검색/대여/예약/출력

```
switch (selectNum) {

// 1. 검색/대여-----
case 1:
    mem.searchItemInfo();
    break;

// 2. 로그인-----
case 2:
    adm.login();

    break;
// 3. 회원가입/이용안내-----
case 3:

    System.out.println("1. 이용안내 | 2. 회원가입");
    selectNum = adm.sc.nextInt();
    adm.sc.nextLine();
    switch (selectNum) {
    case 1:
        mem.showGuide();
        break;
    case 2:
        adm.addInfo();
        break;
    }
    break;

// 4. 내 대여내역-----
case 4:
    if (adm.loginCheck()) {
        mem.returnExtends();
        break;
    } else {
        System.out.println("로그인 후 이용가능합니다.");
        System.out.println("먼저 로그인 해주세요.");

        continue;
    }

// 5.로그아웃-----
case 5:
    adm.logout();
    break;
```

```
-----
대   여   대   여   L I B R A R Y   -----
-----

1. 검색/대여 | 2.로그인 | 3.이용안내/회원가입 | 4.내 대여내역(반납/연장) | 5.로그아웃 | 6.관리자페이지 | 7.프로그램 종료
-----

3
1. 이용안내 | 2.회원가입
1
=====
                               이       용       안       내
=====

1. 비회원은 대여 검색을 하여 대여 가능한 목록을 볼수 있으며, 대여 하기위해서는 회원가입을 해야지만 대여 가능합니다.
2. 회원가입은 이름, 나이, 전화번호, 주소, 이메일 그리고 아이디와 패스워드 입력을 통해서 회원가입이 합니다.
3. 회원가입이 완료 되었으면 로그인을 하고 도서, DVD, 게임 중 목록을 보고 선택하여 대여를 할수 있습니다.
4. 개인회원은 도서, DVD, 게임 품목은 기본정보와 대여가능횟수, 관심수, 재고, 반납예정일을 화면상에서 확인 할수 있습니다.

-----

대여자료 : 도서 / DVD / 게임
대여가능 자료수 : id당 5개
대여기간 : 자료당 7일

연장가능 횟수 : id당 1회
연장가능 일수 : 7일
연체시 연체일수만큼 대여불가
=====
```



## 기능별 소개

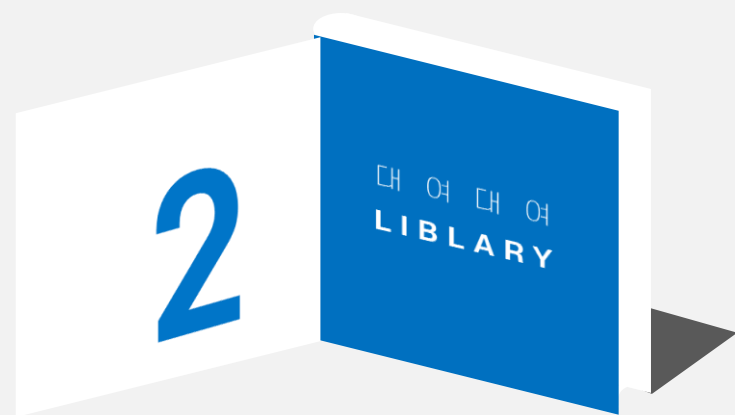
### 1) 메인 \_검색/대여/예약/출력

```
// 6. 관리자페이지-----
case 6:
    if(adm.AdminLogin()) {
        System.out.println("너무 많이 틀리셨습니다. 메인 페이지로 돌아갑니다.");
        break;
    } else {
        System.out.println("관리자로 로그인 하였습니다.");
    }
    while (true) {
        System.out.println("1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃");
        try {
            selectNum = adm.sc.nextInt();
            adm.sc.nextLine();
        } catch (Exception e) {
            System.out.println("잘못된 입력입니다.\n1~6사이의 숫자를 입력해주세요.");
            adm.sc.nextLine();
            continue;
        }
        switch (selectNum) {
            case 1:
                RentalMenu.memeberView();
                continue;
            case 2:
                RentalMenu.bookView();
                continue;
            case 3:
                RentalMenu.dvdView();
                continue;
            case 4:
                RentalMenu.gameView();
                continue;
            case 5:
                RentalMenu.rentalListView();
                continue;
            case 6:
                break;
            default:
                System.out.println("잘못된 입력입니다.\n1~6사이의 숫자를 입력해주세요.");
                continue;
        }
        break;
    } // while end
    continue;
case 7:
    System.out.println("프로그램이 종료됩니다.");
    System.exit(0);
default:
    System.out.println("잘못된 입력입니다.\n1~7사이 숫자를 입력해 주세요.");
    continue;
```

```
-----
대   여   대   여   L I B R A R Y
-----

1. 검색/대여 | 2.로그인 | 3.이용안내/회원가입 | 4.내 대여내역(반납/연장) | 5.로그아웃 | 6.관리자페이지 | 7.프로그램 종료

6
-----
관리자페이지 입니다.
비밀번호를 입력해주세요...>>
-----
0000
관리자로 로그인 하였습니다.
1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃
1
회원관리페이지입니다.
1.회원 상세정보 | 2.회원정보 수정 | 3.회원정보 삭제 | 4. 뒤로가기
```



## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

\_박명훈

회원, 자료, 대여내역 정보추가

### 3) 관리자매니저

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저

검색 / 대여 / 예약 / 반납 / 연장 / 연체



```
public class MemberInfo {

    public boolean loginCheck; //로그인
    public String name; //이름
    public int age; //나이
    public String phoneNum; //전화번호
    public String addr; //주소
    public String email; //이메일

    //인스턴스 변수 초기화
    public MemberInfo(String name,int age, String phoneNum,String addr,String email){
        this.name = name;
        this.age = age;
        this.phoneNum = phoneNum;
        this.addr = addr;
        this.email = email;
        this.loginCheck = false;
    }

    public MemberInfo(){

    }

    void showBasicInfo() {
        System.out.println("이름 : "+name+"\t\t나이 : "+age+"\t\t\t\t\t전화번호 : "+phoneNum+"\t\t주소 : "+addr+"\t\t이메일 : "+email);
    }

    void showAllInfo() {

    }

    void AllInformation() {

    }

}

}
```

관리자로 로그인 하였습니다.

1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃

1  
회원관리페이지입니다.

1. 회원 상세정보 | 2. 회원정보 수정 | 3. 회원정보 삭제 | 4. 뒤로가기

이름 : 박명훈 등급 : 2 아이디 : hoon	나이 : 41 대여권수 : 0	전화번호 : 010-1234-3210 대여가능권수 : 5	주소 : 서울시 종로구 송인동 연장가능횟수 : 1	이메일 : hoon@naver.com 연체 : 0
이름 : 박성민 등급 : 3 아이디 : min	나이 : 12 대여권수 : 0	전화번호 : 010-8080-9632 대여가능권수 : 5	주소 : 서울시 종로구 혜화동 연장가능횟수 : 1	이메일 : min@naver.com 연체 : 0
이름 : 박혜미 등급 : 1 아이디 : mi	나이 : 20 대여권수 : 0	전화번호 : 010-6372-0908 대여가능권수 : 5	주소 : 서울시 종로구 가회동 연장가능횟수 : 1	이메일 : mi@naver.com 연체 : 0
이름 : 임재영 등급 : 3 아이디 : lim	나이 : 28 대여권수 : 0	전화번호 : 010-6611-4511 대여가능권수 : 5	주소 : 서울시 종로구 창신2동 연장가능횟수 : 1	이메일 : lim@naver.com 연체 : 0
이름 : 임재영 등급 : 2 아이디 : young	나이 : 16 대여권수 : 0	전화번호 : 010-8012-7777 대여가능권수 : 5	주소 : 서울시 종로구 창신2동 연장가능횟수 : 1	이메일 : young@naver.com 연체 : 0

#### 4.checkPw : 비밀번호 비교 메서드 생성

기능별 소개

2) 회원정보/ 자료정보/ 대여정보

Admin() : 관리자 클래스

```
package master;

public class Admin{

    boolean loginCheck;           //로그인
    private String adminId;        //관리자 아이디
    private String adminPw;        //관리자 비밀번호

    Admin(String adminId,String adminPw){

        this.adminId = adminId;
        this.adminPw = adminPw;
        this.loginCheck = false;

    }

    boolean checkId(String adminId) {
        return this.adminId.equals(adminId);
    }

    boolean checkpw(String adminPw) {
        return this.adminPw.equals(adminPw);
    }

}
```

1. 관리자 로그인 아이디와 패스워드 클래스 생성 후 생성자로 통한 인스턴스 변수 초기화
2. checkId : 아이디 비교 메서드 생성
3. checkPw : 비밀번호 비교 메서드 생성

RentalItemInfo() : 자료정보 클래스

```
public class RentalItemInfo {

    public String title;           //자료명
    public String genre;           //장르
    public String localData;       //자료위치
    public String rentInfo;        //대여상태
    public int numOfItem;          //자료의 개수
    public int rentalCount;        //자료의 대여횟수

    //인스턴스 변수 초기화
    public RentalItemInfo(String title,String genre,String localData){
        this.title = title;
        this.genre = genre;
        this.localData = localData;
        this.rentInfo = "대여가능" ;
        this.numOfItem = 1;
        this.rentalCount = 0;
    }

    public RentalItemInfo() {
        // TODO Auto-generated constructor stub
    }

    //기본정보 출력 메서드
    void showBasicInfo() {
        System.out.println("자료명 : "+title+"\t장르 : "+genre+"\t자료위치 : "+localData);
    }

    void showAllInfo() {

    }

    boolean checkTitle(String title) {
        return this.title.equals(title);
    }

}
```

Game()

```
public class Game extends RentalItemInfo{

    public String producer;        //제작사
    public int limitAge;           //제한연령
    public String story;           //스토리
    public String launchDate;       //출시일
    public int rentalCount;         //대여횟수
    public int like;               //좋아요
    public String returnDate;       //반납예정일

    //인스턴스 변수 초기화
    public Game(String title, String genre, String localData,String producer,int limitAge,String story,String launchDate) {
        super(title, genre, localData);
        this.producer = producer;
        this.limitAge = limitAge;
        this.story = story;
        this.launchDate = launchDate;
        this.rentalCount = 0;
        this.like = 0;
        this.returnDate = null;
    }

    public Game(){

    }

    @Override
    void showBasicInfo() {
        super.showBasicInfo();
        System.out.println("제작사 : "+producer);
    }

    @Override
    void showAllInfo() {
        super.showBasicInfo();
        System.out.println("제한연령 : "+limitAge+"\t출시일 : "+launchDate+"\t판매량 : "+story);
        System.out.println("대여횟수 : "+rentalCount+"\t좋아요 : "+like);
        System.out.println("반납예정일 : "+returnDate);
    }

    @Override
    boolean checkTitle(String title) {
        return super.checkTitle(title);
    }

}
```

Book()

```
public class Book extends RentalItemInfo{

    public String author;          //작가
    public int limitAge;           //제한연령
    public String story;           //스토리
    public String launchDate;       //출시일
    public int rentalCount;         //대여횟수
    public int like;               //좋아요
    public String returnDate;       //반납예정일

    //인스턴스 변수 초기화
    public Book(String title, String genre, String localData,String author,int limitAge,String story,String launchDate) {
        super(title, genre, localData);
        this.author = author;
        this.limitAge = limitAge;
        this.story = story;
        this.launchDate = launchDate;
        this.rentalCount = 0;
        this.like = 0;
        this.returnDate = null;
    }

    public Book(){

    }

    @Override
    void showBasicInfo() {
        super.showBasicInfo();
        System.out.println("작가 : "+author);
    }

    @Override
    void showAllInfo() {
        super.showBasicInfo();
        System.out.println("제한연령 : "+limitAge+"\t출시일 : "+launchDate+"\t판매량 : "+story);
        System.out.println("대여횟수 : "+rentalCount+"\t좋아요 : "+like);
        System.out.println("반납예정일 : "+returnDate);
    }

    @Override
    boolean checkTitle(String title) {
        return super.checkTitle(title);
    }

}
```

DVD()

```
public class DVD extends RentalItemInfo{

    public String format;          //포맷
    public String rating;          //등급
    public int limitAge;           //제한연령
    public String story;           //스토리
    public String launchDate;       //출시일
    public int rentalCount;         //대여횟수
    public int like;               //좋아요
    public String returnDate;       //반납예정일

    //인스턴스 변수 초기화
    public DVD(String title, String genre, String localData,String format,String rating,int limitAge,String story,String launchDate) {
        super(title, genre, localData);
        this.format = format;
        this.rating = rating;
        this.limitAge = limitAge;
        this.story = story;
        this.launchDate = launchDate;
        this.rentalCount = 0;
        this.like = 0;
        this.returnDate = null;
    }

    public DVD(){

    }

    @Override
    void showBasicInfo() {
        super.showBasicInfo();
        System.out.println("포맷 : "+format);
    }

    @Override
    void showAllInfo() {
        super.showBasicInfo();
        System.out.println("제한연령 : "+limitAge+"\t출시일 : "+launchDate+"\t판매량 : "+story);
        System.out.println("대여횟수 : "+rentalCount+"\t좋아요 : "+like);
        System.out.println("반납예정일 : "+returnDate);
    }

    @Override
    boolean checkTitle(String title) {
        return super.checkTitle(title);
    }

}
```

1. RentalItemInfo : 자료정보 클래스
2. Game, Book, DVD : RentalItemInfo 상속 후 기본정보 추가  
추가 정보는 위와 회원정보와 동일





## 기능별 소개

### 2) 회원정보/ 자료정보/ 대여정보

#### showBasicInfo() : Book 클래스 기본 실행문

1.도서 | 2.DVD | 3. 게임  
1

책 기본 정보 출력

자료명 : 더 해빙      장르 : 자기개발      재고 : 5      자료위치 : A-1  
저 자 : 이서문,홍주연

자료명 : 자리의 힘      장르 : 역사      재고 : 5      자료위치 : A-2  
저 자 : 팀 마샬

#### showAllInfo() : Book 클래스 상세 실행문

연컨택트

자료명 : 연컨택트      장르 : 경제/경영      재고 : 5      자료위치 : A-4  
제한연령 : 0      출판일 : 2020.04.20

설명 : 코로나19 이후 우리 삶은 어떻게 달라져 있을까? 2020년 코로나19로 촉발된 전혀 새로운 삶의 방식 ‘연컨택트’! 불편한 소통보다 ‘편리한 단절’을 꿈꾸는 현대인의 욕망, 라이프스타일의 거대한 진화는 이미 시작됐다! 접촉 불안이 가져온 일상의 대 전환기! 불안과 위험의 시대를 건너는 우리의 자세는 어떻게 달라져야 하는가? 2020년 코로나19로 인해 세상의 관심은 ‘..  
대여횟수 : 0      관심수 : 0

#### showBasicInfo() : Game 클래스 기본 실행문

게임 기본 정보 출력

자료명 : 젤다의 전설      장르 : 오픈에어 어드벤처      재고 : 5      자료위치 : G-1  
제작사 : 닌텐도

자료명 : 슈퍼마리오      장르 : 모험      재고 : 5      자료위치 : G-2  
제작사 : 닌텐도

자료명 : 포켓몬스터      장르 : 풀플레이      재고 : 5      자료위치 : G-3  
제작사 : 닌텐도

#### showAllInfo() : Game 클래스 상세 실행문

앞보시는 Game명을 입력해주세요.

포켓몬스터

1.대여 | 2.예약

자료명 : 포켓몬스터      장르 : 풀플레이      재고 : 5      자료위치 : G-3  
제한연령 : 0      출시일 : 2019.11.15

설명 : 《포켓몬스터 소드·실드》는 게임프릭스 제작, 포켓몬 컴퍼니 및 닌텐도 배급의 풀플레이 게임이다.  
닌텐도 스위치 전용. 포켓몬 연작의 제8세대를 여는 작품이며 더불어 《레츠고! 피카츄·이브이》의 다음가는 프랜차이즈의 두 번째 가정용 콘솔 출시작이다.  
대여횟수 : 0      관심수 : 0

#### showBasicInfo() : DVD 클래스 기본 실행문

DVD 기본 정보 출력

자료명 : 레이니 데이 인 뉴욕      장르 : 멜로/로맨스      재고 : 5      자료위치 : B-1  
감독 : 무디 앨런

자료명 : 트롤:월드 투어      장르 : 애니메이션      재고 : 5      자료위치 : B-2  
감독 : 월트토른데이빗P.스미스

자료명 : 저 산 너머      장르 : 드라마      재고 : 5      자료위치 : B-3  
감독 : 최종태

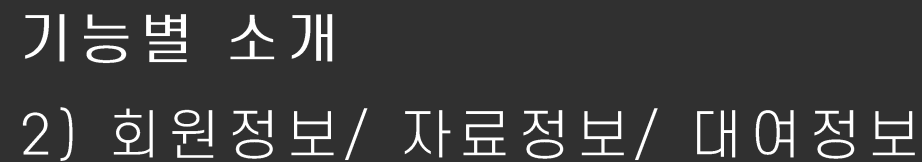
#### showAllInfo() : DVD 클래스 상세 실행문

프리즈 이스케이프

1.대여 | 2.예약

자료명 : 프리즌 이스케이프      장르 : 모험/스릴러      재고 : 5      자료위치 : B-5  
제한연령 : 0      출판일 : 2020.05.06      상영시간 : 106분

설명 : 404일의 감금, 나가야 할 문은 15개! 성공률 0%의 탈옥이 시작된다! 인권운동가 ‘팀’과 ‘스티븐’은 억울한 판결로 투옥된다. 둘은 불의에 굴복하지 않고 탈출을 결심한다. 나가기 위해 열어야 할 강철 문은 15개! 그들은 나뭇조각으로 열쇠를 만들기 시작하고, 지금껏 아무도 성공한 적 없는 0%의 확률 속에서 목숨을 건 단 한 번의 기회를 노리는데...  
대여횟수 : 0      관심수 : 0



```

import master.Member;

public class MemberData extends Member {

    public MemberData() {
        this.name = "User";
        this.age = 21;
        this.phoneNum = "010-8888-8888";
        this.addr = "Halla 822 B2C";
        this.email = "qwasad@naver.com";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.level = 1;
        this.memberType = 1;
        this.rentalLevel = 1;
        this.memberStatus = 1;
        this.verDate = 1;
        this.rentalDate = 1;
        this.rentalStatus = 1;
        this.rentInfo = "US";
        this.dataStatus = 1;
    }
}

import master.Member;

public class MemberData extends Member {

    public MemberData() {
        this.name = "User";
        this.age = 21;
        this.phoneNum = "010-8888-8888";
        this.addr = "Halla 822 B2C";
        this.email = "qwasad@naver.com";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.level = 1;
        this.memberType = 1;
        this.rentalLevel = 1;
        this.memberStatus = 1;
        this.verDate = 1;
        this.rentalDate = 1;
        this.rentalStatus = 1;
        this.rentInfo = "US";
        this.dataStatus = 1;
    }
}

import master.Member;

public class MemberData extends Member {

    public MemberData() {
        this.name = "User";
        this.age = 21;
        this.phoneNum = "010-8888-8888";
        this.addr = "Halla 822 B2C";
        this.email = "qwasad@naver.com";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.level = 1;
        this.memberType = 1;
        this.rentalLevel = 1;
        this.memberStatus = 1;
        this.verDate = 1;
        this.rentalDate = 1;
        this.rentalStatus = 1;
        this.rentInfo = "US";
        this.dataStatus = 1;
    }
}

import master.Member;

public class MemberData extends Member {

    public MemberData() {
        this.name = "User";
        this.age = 21;
        this.phoneNum = "010-8888-8888";
        this.addr = "Halla 822 B2C";
        this.email = "qwasad@naver.com";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.phoneNum = "010-8888-8888";
        this.level = 1;
        this.memberType = 1;
        this.rentalLevel = 1;
        this.memberStatus = 1;
        this.verDate = 1;
        this.rentalDate = 1;
        this.rentalStatus = 1;
        this.rentInfo = "US";
        this.dataStatus = 1;
    }
}

```

```

import master.RentalList;

public class UserList extends RentalList {

    String id; // 아이디
    String title; // 제목
    String rentalInfo; // 대여정보
    String rentalDate; // 대여날짜
    String returnDate; // 반납날짜
    String returnLimit; // 반납제한
    String reserveDate; // 예약날짜
    String reserveId; // 예약ID
    String extendDate; // 연장날짜
    String overdue; // 연착료

    public UserList(){}

    this.id = "U00001"; // 예약ID
    this.title = "H" // 예약 제목
    this.rentalInfo = "2020-05-10"; // 대여정보
    this.rentalDate = "2020-05-10"; // 대여날짜
    this.returnDate = "2020-05-15"; // 반납날짜
    this.reserveDate = "2020-05-10"; // 예약날짜
    this.reserveId = "1"; // 예약ID
    this.extendDate = "1"; // 연장날짜
    this.overdue = "1"; // 연착료

}

}

import master.RentalList;

public class UserList1 extends RentalList {

    String id; // 아이디
    String title; // 제목
    String rentalInfo; // 대여정보
    String rentalDate; // 대여날짜
    String returnDate; // 반납날짜
    String returnLimit; // 반납제한
    String reserveDate; // 예약날짜
    String reserveId; // 예약ID
    String extendDate; // 연장날짜
    String overdue; // 연착료

    public UserList1(){}

    this.id = "U00001"; // 예약ID
    this.title = "H" // 예약 제목
    this.rentalInfo = "2020-05-10"; // 대여정보
    this.rentalDate = "2020-05-10"; // 대여날짜
    this.returnDate = "2020-05-15"; // 반납날짜
    this.reserveDate = "2020-05-10"; // 예약날짜
    this.reserveId = "1"; // 예약ID
    this.extendDate = "1"; // 연장날짜
    this.overdue = "1"; // 연착료

}

}

import master.RentalList;

public class UserList1 extends RentalList {

    String id; // 아이디
    String title; // 제목
    String rentalInfo; // 대여정보
    String rentalDate; // 대여날짜
    String returnDate; // 반납날짜
    String returnLimit; // 반납제한
    String reserveDate; // 예약날짜
    String reserveId; // 예약ID
    String extendDate; // 연장날짜
    String overdue; // 연착료

    public UserList1(){}

    this.id = "U00001"; // 예약ID
    this.title = "H" // 예약 제목
    this.rentalInfo = "2020-05-10"; // 대여정보
    this.rentalDate = "2020-05-10"; // 대여날짜
    this.returnDate = "2020-05-15"; // 반납날짜
    this.reserveDate = "2020-05-10"; // 예약날짜
    this.reserveId = "1"; // 예약ID
    this.extendDate = "1"; // 연장날짜
    this.overdue = "1"; // 연착료

}

}

import master.RentalList;

public class UserList15 extends RentalList {

    String id; // 아이디
    String title; // 제목
    String rentalInfo; // 대여정보
    String rentalDate; // 대여날짜
    String returnDate; // 반납날짜
    String returnLimit; // 반납제한
    String reserveDate; // 예약날짜
    String reserveId; // 예약ID
    String extendDate; // 연장날짜
    String overdue; // 연착료

    public UserList15(){}

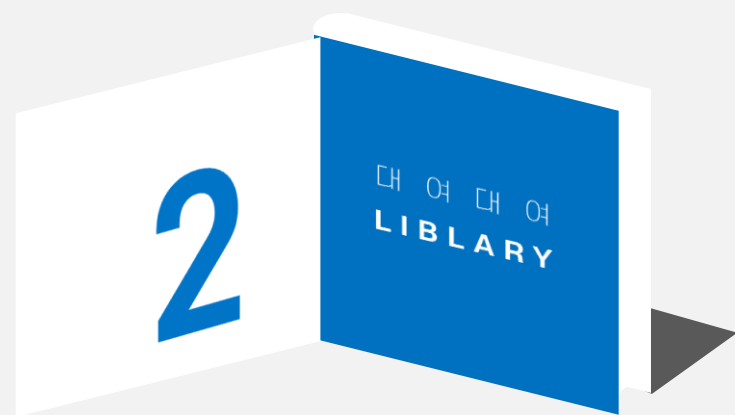
    this.id = "U00015"; // 예약ID
    this.title = "H" // 예약 제목
    this.rentalInfo = "2020-05-10"; // 대여정보
    this.rentalDate = "2020-05-10"; // 대여날짜
    this.returnDate = "2020-05-15"; // 반납날짜
    this.reserveDate = "2020-05-10"; // 예약날짜
    this.reserveId = "1"; // 예약ID
    this.extendDate = "1"; // 연장날짜
    this.overdue = "1"; // 연착료

}

}

```





## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

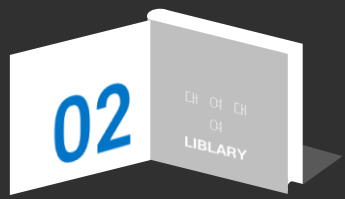
회원, 자료, 대여내역 정보추가

### 3) 관리자매니저 \_박성민

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저

검색 / 대여 / 예약 / 반납 / 연장 / 연체



## 기능별 소개

### 3) 관리자 매니저 \_회원/자료/대여내역관리

사용하실 아이디를 입력해 주세요.(숫자와 영문만 입력가능합니다.)

공백을 입력하셨습니다. 다시 입력해주세요.

*we*

비밀번호를 입력해 주세요.(숫자와 영문만 가능합니다.)

*123*

다시한번 입력해 주세요.(숫자와 영문만 가능합니다.)

*12*

입력하신 비밀번호가 다릅니다. 다시 입력해 주세요.

비밀번호를 입력해 주세요.(숫자와 영문만 가능합니다.)

*123*

다시한번 입력해 주세요.(숫자와 영문만 가능합니다.)

*123*

이름을 입력해 주세요.(한글만 입력 가능합니다)

*박성민1*

한글만 입력가능합니다.

*박성민*

나이를 입력 해주세요.(숫자만 입력 가능합니다.)

*12*

전화번호를 입력해 주세요.(전화번호형식 000-0000-0000 입력만 가능합니다.)

*01012345678*

전화번호 형식 (000-0000-0000) 으로 입력해 주세요.

*010-3150-3150*

주소를 입력해 주세요. (숫자와 한글만 입력가능합니다.)

*대구시*

이메일을 입력해 주세요.(이메일형식 xxx@xxx.xxx식으로 입력해주세요.)

*min@naver*

이메일 형식(xxx@xxx.xxx)으로 입력하여 주세요.

*min@naver.com*

회원가입이 정상적으로 완료되었습니다.

```
//비밀번호 확인 루프를 위한 변수
boolean checkPw = true;

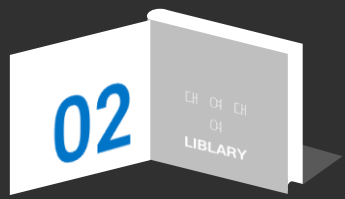
System.out.println("회원가입을 시작합니다.");
System.out.println("사용하실 아이디를 입력해 주세요.(숫자와 영문만 입력가능합니다.);");
id = checkEnN();

while (checkPw) {
    System.out.println("비밀번호를 입력해 주세요.(숫자와 영문만 가능합니다.);");
    pw = checkEnN();
    System.out.println("다시한번 입력해 주세요.(숫자와 영문만 가능합니다.);");
    pw2 = checkEnN();
    if (pw.equals(pw2)) {
        System.out.println("이름을 입력해 주세요.(한글만 입력 가능합니다.);");
        name = checkKorean();

        while (true) {
            System.out.println("나이를 입력 해주세요.(숫자만 입력 가능합니다.);");
            try {
                age = sc.nextInt();
                sc.nextLine();
            } catch (InputMismatchException e) {
                System.out.println("숫자를 입력해 주세요.");
                sc.nextLine();
                continue;
            }
        }
    }
}
```

#### 회원 추가 (회원가입)

- 익숙한 회원가입
- 각각 구문에 적절한 예외처리 및 정규표현식을 사용하여 원하는 값만 입력할 수 있게 제작



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

```
System.out.println("수정하실 분의 아이디를 선택해 주세요.");  
String id = checkVal();  
  
int index = searchIndex(id);
```

```
System.out.println("찾으시는 아이디가 없습니다.");
```

회원관리페이지입니다.

1.회원 상세정보 | 2.회원정보 수정 | 3.회원정보 삭제 | 4. 뒤로가기

2

수정하실 분의 아이디를 선택해 주세요.

min

1. 비밀번호 | 2. 전화번호 | 3.주소 | 4. 이메일 |5. 뒤로가기

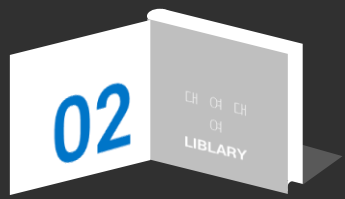
2

변경하실 전화번호를 입력해 주세요.(xxx-xxx(x)-xxxx 형식으로 가능)

|

## 회원 정보수정

- 아이디를 기준으로 회원을 검색
- 찾는 아이디가 없을시 그에 맞는 출력문을 실행
- 아이디가 있을시 변경하고자 하는 정보를 각각 변경 가능하게 함
- 각각 구문에 적절한 예외처리 및 정규표현식을 사용하여 원하는 값만 입력할 수 있게 제작



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

```
//회원정보보기
//객체 전체 목록 출력
public void showInfo() {
    for (int i = 0; i < member.size(); i++) {
        member.get(i).showAllInfo();
        System.out.println("-----");
    }
}
```

```
//회원정보 삭제
public void deleteInfo() {
    System.out.println("삭제하실분의 아이디를 입력해 주세요.");
    String id = checkVal();

    int index = searchIndex(id);
    if (index < 0) {
        System.out.println("삭제하고자하는 아이디가 존재하지 않습니다");
    } else {
        System.out.println("아이디가 삭제되었습니다.");
        member.remove(index);
    }
}
```

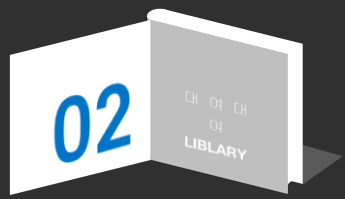
#### 회원 정보삭제/ 전체정보 출력

##### 전체정보 출력

- for문을 사용하여 전체 회원 정보를 출력

##### 회원정보 삭제

- 삭제할 회원의 아이디를 입력 받아 아이디가 있을시 삭제 없을시 적절한 출력문을 출력



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

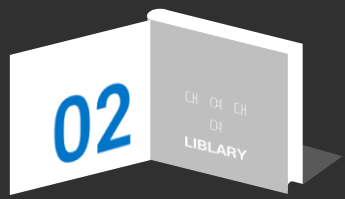
```
//회원 로그인
public void login() {

    if (loginCheck() == false) {
        boolean check = true;
        int loginCnt = 0;
        int pwCnt = 0;
        while (check) {
            System.out.println("아이디를 입력해 주세요.");
            String id = checkVal();

            int index = searchIndex(id);
            if (index >= 0) {
                System.out.println("비밀번호를 입력해 주세요.");
                String pw = checkVal();
                if (member.get(index).getPw().equals(pw)) {
                    System.out.println("로그인이 완료되었습니다.");
                    member.get(index).loginCheck = true;
                    //System.out.println(member.get(index).loginCheck);
                    check = false;
                    break;
                } else {
                    System.out.println("비밀번호를 다시 입력해 주세요.");
                    pwCnt++;
                    if(pwCnt == 3){
                        System.out.println("3회 잘못 입력 하셨습니다. 메인으로 돌아갑니다.");
                        break;
                    }
                    continue;
                }
            } else {
                System.out.println("입력하신 아이디가 없습니다. 다시 입력해주세요.");
                loginCnt++;
                if(loginCnt == 3){
                    System.out.println("3회 잘못 입력 하셨습니다. 메인으로 돌아갑니다.");
                    break;
                }
                continue;
            }
        }
    } else {
        System.out.println("이미 로그인 중입니다.");
    }
}
```

#### 회원 로그인

- loginCheck()메소드를 이용하여 로그인 중인지 아닌지를 판단함
- 로그인 중일 시 적절한 출력문 출력
- 비로그인일 시 로그인 절차를 진행
- 아이디 or 비밀번호 3회 오타시 메인화면으로 돌아가게 하여 무한루핑 방지



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

```
//관리자 로그인
public boolean AdminLogin() {
    String loginPw = "0000";
    boolean loginCheck = true;
    int failCnt = 0;
    System.out.println("-----");
    System.out.println("관리자페이지 입니다.");

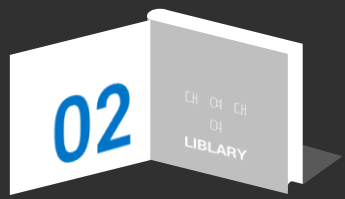
    while (loginCheck) {

        System.out.println("비밀번호를 입력해주세요..>>");
        System.out.println("-----");
        String adminPw = sc.nextLine();

        if (loginPw.equals(adminPw)) {
            loginCheck = false;
            break;
        } else {
            System.out.println("비밀번호가 틀립니다. 다시 입력해 주세요.");
            failCnt++;
            if (failCnt == 3) {
                break;
            }
            continue;
        }
    }
    return loginCheck;
}
```

#### 관리자 로그인

- 관리자는 객체 생성을 할 필요 없이 초기 비밀번호를 지정하여 접속 가능하게 함
- 입력받은 비밀번호가 일치할 시 관리자 페이지로 이동
- 로그아웃의 경우 메인 화면으로 나가게 되면 자동 로그아웃



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

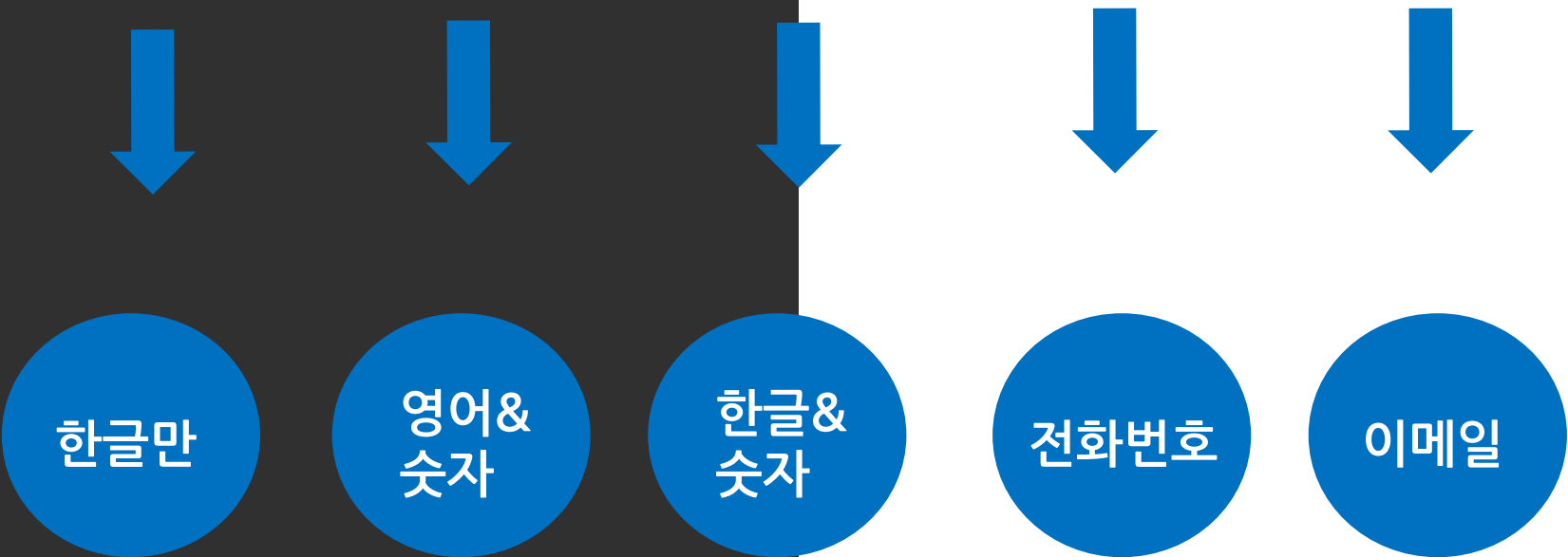
```
public void logOut() {  
    if (loginCheck()) {  
        for (int i = 0; i < member.size(); i++) {  
            if (member.get(i).loginCheck == true) {  
                member.get(i).loginCheck = false;  
                System.out.println("로그아웃 되었습니다.");  
                break;  
            }  
        }  
    } else {  
        System.out.println("로그인된 계정이 없습니다.");  
    }  
}
```

#### 회원 로그아웃

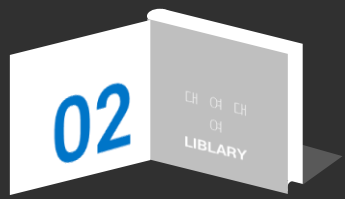
- 회원배열을 검색하여 로그인체크 값이 true인 회원의 값을 false로 변경한 후 적절한 출력문을 출력하여 로그아웃이 가능하게 함

체크 메서드

```
//공백 체크 메소드(String)
String checkVal() {
    String checkMsg = null;
    while (true) {
        checkMsg = sc.nextLine();
        String p = "^\\s*$";
        boolean m = Pattern.matches(p, checkMsg);
        if (m) {
            System.out.println("공백을 입력하셨습니다. 다시 입력해주세요.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```







## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

#### 체크 메서드

```
//한글만 입력받는 메소드
String checkKorean() {
    String checkMsg = null;
    while (true) {
        checkMsg = checkVal();
        String p = "[가-힣]*";
        boolean m = Pattern.matches(p, checkMsg);
        if (m==false) {
            System.out.println("한글만 입력가능합니다.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```

```
//이메일 입력받는 메소드
String checkEmail() {
    String checkMsg = null;
    while (true) {
        checkMsg = checkVal();
        String p = "[a-zA-Z0-9]+@[a-zA-Z0-9]+\\.\\.[a-zA-Z]{2,6}";
        boolean m = Pattern.matches(p, checkMsg);
        if (m == false) {
            System.out.println("이메일 형식 (xxx@xxx.xxx)으로 입력하여 주세요.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```

```
//한글과 숫자만 받는 메소드
String checkKnN() {
    String checkMsg = null;
    while (true) {
        checkMsg = checkVal();
        String p = "[가-힣0-9]*";
        boolean m = Pattern.matches(p, checkMsg);
        if (m==false) {
            System.out.println("한글과 숫자의 조합만 입력가능합니다.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```

```
//영어와 숫자만 받는 메소드(id)
String checkEnN() {
    String checkMsg = null;
    while (true) {
        checkMsg = checkVal();
        String p = "[a-zA-Z0-9]*";
        boolean m = Pattern.matches(p, checkMsg);
        if (m==false) {
            System.out.println("영어와 숫자의 조합만 입력가능합니다.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```

```
//전화번호 입력받는 메소드
String checkPhoneNum() {
    String checkMsg = null;
    while (true) {
        checkMsg = checkVal();
        String p = "01(?:0|1|[6-9])-(?:\\d{3}|\\d{4})-\\d{4}";
        boolean m = Pattern.matches(p, checkMsg);
        if (m == false) {
            System.out.println("전화번호 형식 (000-0000-0000) 으로 입력해 주세요.");
            continue;
        } else {
            break;
        }
    }
    return checkMsg;
}
```

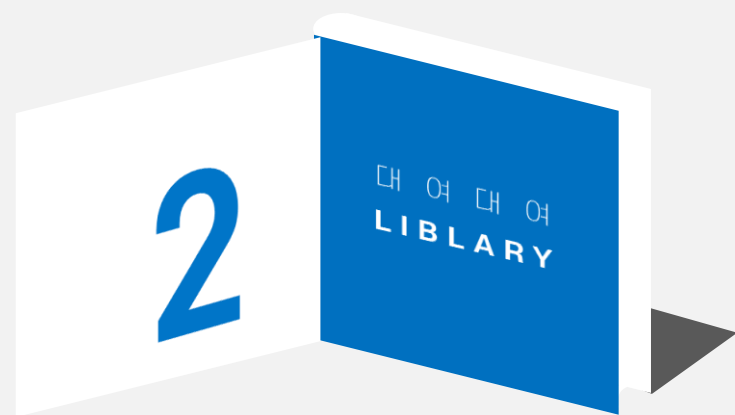
### 체크 메서드 - 로그인체크/ 각종검색에 쓰인 메서드들

```
// 로그인체크해서 로그인상태인지 아닌지 반환
boolean loginCheck() {
    boolean loginCheck = false;
    for (int i = 0; i < this.member.size(); i++) {
        if ((this.member.get(i).loginCheck) == true) {
            loginCheck = true;
            break;
        }
    }
    return loginCheck;
}

// 로그인체크해서 인덱스 반환
int loginCheckIndex() {
    int loginCheckIndex = -1;
    for (int i = 0; i < this.member.size(); i++) {
        if (this.member.get(i).loginCheck == true) {
            loginCheckIndex = i;
            break;
        }
    }
    return loginCheckIndex;
}
```

```
int searchIndex(String id) {
    //정상적인 index 값은 0~이상의 값, 찾지 못했을 때 구분 값 -1을 사용
    int searchIndex = -1;
    //배열의 반복으로 id값을 비교해서 index 값을 찾는다.
    for (int i = 0; i < member.size(); i++) {
        if (member.get(i).checkId(id)) {
            searchIndex = i;
            break;
        }
    }
    return searchIndex;
}
```





## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

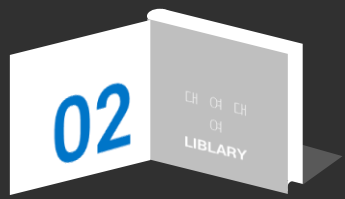
회원, 자료, 대여내역 정보추가

### 3) 관리자매니저 \_임재영<sup>96</sup>

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저

검색 / 대여 / 예약 / 반납 / 연장 / 연체



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

//아이템배열 선언

```
private ArrayList<Book> books;  
private ArrayList<DVD> dvd;  
private ArrayList<Game> game;
```

아이템 배열 선언  
book,dvd,game

Book,DVD,Game클래스를 배열 선언

```
private AdminManager() {  
    this.member = new ArrayList<>();  
    this.books = new ArrayList<>();  
    this.dvd = new ArrayList<>();  
    this.game = new ArrayList<>();  
    sc = new Scanner(System.in);  
}
```

아이템 배열 초기화  
book,dvd,game

생성자에서 선언한 배열 초기화

//book데이터 객체

```
Bookdata book1 = new Bookdata();  
Bookdata2 book2 = new Bookdata2();  
Bookdata3 book3 = new Bookdata3();  
Bookdata4 book4 = new Bookdata4();  
Bookdata5 book5 = new Bookdata5();
```

아이템 데이터 배열 선언  
book,dvd,game  
EX)book

data패키지 내의 데이터 기본값 선언

```
this.books.add(book1);  
this.books.add(book2);  
this.books.add(book3);  
this.books.add(book4);  
this.books.add(book5);
```

AdminManager싱글톤 구현  
및 getter생성

한 번의 객체생성으로 재사용  
가능, 메모리 낭비를 방지토록  
싱글톤으로 구현

//어드민매니저 객체 싱글톤 구현

```
private static AdminManager manager = new AdminManager();  
  
public static AdminManager getInstance() {  
    return manager;  
}  
  
public ArrayList<Member> getMember() {  
    return member;  
}  
  
public ArrayList<Book> getBooks() {  
    return books;  
}  
  
public ArrayList<DVD> getDvd() {  
    return dvd;  
}  
  
public ArrayList<Game> getGame() {  
    return game;  
}
```

# 아이템 전체정보 및 기본정보 출력메서드

```
//자료관리
//도서 정보보기
public void showBookInfo() {
    System.out.println("책 전체정보 출력");
    for (int i = 0; i < books.size(); i++) {
        books.get(i).showAllinfo();
        System.out.println("-----");
    }
}

public void showBookBasic() {
    System.out.println("책 기본 정보 출력");
    for (int i = 0; i < books.size(); i++) {
        books.get(i).showBasicInfo();
        System.out.println("-----");
    }
}
```

아이템 정보 출력메서드

-아이템 객체 내 저장되어 있는 get(i)의0번째 index부터 아이템 index의 사이즈까지 (EX) books.size) 전체 정보 및 기본 정보 출력  
EX)  
showALLInfo();  
showBasicInfo();

관리자로 로그인 하였습니다.

1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃

2

도서관리 페이지 입니다.

책 전체정보 출력

자료명 : 더 해빙

장르 : 자기개발

재고 : 5

자료위치 : A-1

제한연령 : 0

출판일 : 2020.03.01

설명 : 마음가짐을 다듬으며 원하는 것을 얻는 놀라운 가르침!부와 행운을 만나는 출발점, 마법의 감정 HAVING! 국내 최초로 미국에서 선(先)출간되어 세계가 먼저 찾아 읽은 『더 해빙(THE HAVING)』. 세계적인 부자들이 성공 비결로 꼽았다는 ‘운’. 행운은 타고나는 것일까? 소수에게만 주어지는 것일까? 물려받은 재산도, 특출난 재능도 없는 사람도 부자가 될 수 있을..

대여횟수 : 0

관심수 : 0

-----

자료명 : 자리의 힘

장르 : 역사

재고 : 4

자료위치 : A-2

제한연령 : 0

출판일 : 2016.08.10

설명 : 세계는 왜 ‘자리’를 두고 분쟁하는가?지리라는 렌즈를 통해 세계를 조망한 책 『자리의 힘』. 25년 이상 30개 이상의 분쟁 지역을 직접 현장에서 취재하며 국제 문제 전문 저널리스트로 활동해온 저자가 중국, 미국, 서유럽, 러시아, 한국과 일본, 아프리카, 라틴 아메리카, 중동, 인도와 파키스탄, 북극 등 전 세계를 10개의 지역으로 나눠 ‘자리의 힘’이 21세기 현대사에.

대여횟수 : 1

관심수 : 0

-----



//도서정보삭제

```
public void deleteBookInfo() {
    System.out.println("삭제하고자 하는 책 이름을 선택해주세요");
    String title = sc.nextLine();
    int index = searchBookInfo(title);

    if (index < 0) {
        System.out.println("삭제하고자 하는 책이 존재하지 않습니다.");
    } else {
        System.out.println("선택하신 책이 삭제되었습니다.");
        books.remove(index);
    }
}
```

아이템 정보 삭제 메서드

# 아이템 정보 삭제 메서드

-삭제하고자 하는 아이템 정보를 title(이름)로 검색  
조건문을 사용하여 index가 0보다 작을 시에는 배열에 담겨있지 않다는 메시지를 나타내고 아닐시에는 선택한 아이템 정보를 삭제  
EX) book.remove(index)

삭제하고자 하는 책 이름을 선택해주세요  
가면 상장 사진사건  
|선택하신 책이 삭제되었습니다.  
도서관리 페이지입니다.  
책 전체정보 출력

자료명 : 더 해빙	장르 : 자기개발	재고 : 5	자료위치 : A-1
제한연령 : 0	출판일 : 2020.03.01		
설명 : 마음가짐을 다듬으며 원하는 것을 얻는 놀라운 가르침!부와 행운을 만나는 출발점, 마법의 감정 HAVING! 국내 최초로 미국에서 선(先)출간되어 세계가 먼저 찾아 읽은 『더 해빙 (THE HAVING)』. 세계적인 부자들이 성공 비결로 꼽았다는 ‘운’. 행운은 타고나는 것일까? 소수에게만 주어지는 것일까? 물려받은 재산도, 특출난 재능도 없는 사람도 부자가 될 수 있을..			
대여횟수 : 0	관심수 : 0		

-----

자료명 : 자리의 힘	장르 : 역사	재고 : 4	자료위치 : A-2
제한연령 : 0	출판일 : 2016.08.10		
설명 : 세계는 왜 ‘자리’를 두고 분쟁하는가?지리라는 렌즈를 통해 세계를 조망한 책 『자리의 힘』. 25년 이상 30개 이상의 분쟁 지역을 직접 현장에서 취재하며 국제 문제 전문 저널리스트로 활동해온 저자가 중국, 미국, 서유럽, 러시아, 한국과 일본, 아프리카, 라틴 아메리카, 중동, 인도와 파키스탄, 북극 등 전 세계를 10개의 지역으로 나눠 ‘자리의 힘’이 21세기 현대사에.			
대여횟수 : 1	관심수 : 0		

-----

자료명 : 1cm 다이빙	장르 : 시	재고 : 0	자료위치 : A-3
제한연령 : 0	출판일 : 2020.01.21		
설명 : 텀블벅 펀딩 프로젝트 1,000% 달성! 출간 전부터 난리가 난 책! 우리 사회에는 행복해지기 위해 필요한 것이 참 많습니다. 좋은 대학, 이름난 회사, 괜찮은 연봉, 안정적인 가정. 끝이 아닙니다. 내 자식까지 이어지는 좋은 대학, 이름난 회사, 괜찮은 연봉... 이거... 몇 명이나 가능할까요? 재산이라고 대출금 밖에 없는 서른 살 예비 신랑과, 2년간 집에만 있던 스물여섯 ..			
대여횟수 : 5	관심수 : 0		

-----

자료명 : 5년후 나에게	장르 : 자기개발	재고 : 4	자료위치 : A-4
제한연령 : 0	출판일 : 2019.11.01		
설명 : 5년 후 나에게 삶에서 가장 눈부시게 빛나는 1,825개의 하루를 선물하다!가장 쉽고 지혜로운 방법으로 우리를 깊이 생각하게 하고, 하루하루 성장의 글을 쓰게 하는 『5년 후 나에게 Q&A A DAY』. 하루에 하나씩, 1년 동안 그 답을 기록할 수 있는 지혜롭고 영감에 찬 365개의 질문을 담아 바쁜 일상 속에서 미처 발견하지 못했던 삶을 살아가는 이유, 삶의 진정한 행복과..			
대여횟수 : 1	관심수 : 0		

-----

자료명 : 언컨택트	장르 : 경제/경영	재고 : 3	자료위치 : o
제한연령 : 0	출판일 : 2020.04.20		
설명 : 코로나19 이후 우리 삶은 어떻게 달라져 있을까? 2020년 코로나19로 촉발된 전혀 새로운 삶의 방식 ‘언컨택트’! 불편한 소통보다 ‘편리한 단절’을 꿈꾸는 현대인의 욕망, 라이프스타일의 거대한 진화는 이미 시작됐다! 접촉 불안이 가져온 일상의 대전환기! 불안과 위험의 시대를 건너는 우리의 자세는 어떻게 달라져야 하는가? 2020년 코로나19로 인해 세상의 관심은 ‘...’			
대여횟수 : 2	관심수 : 0		

-----

1. 도서 추가 | 2. 도서수정 | 3. 도서삭제 | 4. 뒤로가기

```
//도서정보 검색
public int searchBookInfo(String title) {
    //정상적인 index 값은 0~이상의 값, 찾지 못했을 때 구분 값 -1을 사용
    int searchIndex = -1;

    //배열의 반복으로 title값을 비교해서 index 값을 찾는다.
    for (int i = 0; i < books.size(); i++) {
        if (books.get(i).title.equals(title)) {
            searchIndex = i;
            break;
        }
    }
    return searchIndex;
}

//도서정보수정
public void editBookInfo() {

    System.out.println("변경하고자하는 책 이름을 입력해주세요");
    String title = sc.nextLine();

    int index = searchBookInfo(title);

    if (index < 0) {
        System.out.println("찾으시는 책의 정보가 존재하지 않습니다.");
    } else {
        String editTitle = books.get(index).title;

        System.out.println("수정 사항을 입력합니다");
        System.out.println("책 이름은: " + editTitle + "입니다");
        System.out.println("자료위치는");
        String localData = sc.nextLine();

        books.get(index).localData = localData;
    }
}
```

자료명 : 가면 산장 살인사건	장르 : 범죄 추리	재고 : 5	자료위치 : A-6
제한연령 : 12	출판일 : 2014.09.26		
설명 : 강도와 인질 간의 숨 막히는 줄다리기!			
대여횟수 : 0	관심수 : 0		

아이템 정보 검색메서드

아이템 정보 수정메서드

수정 후

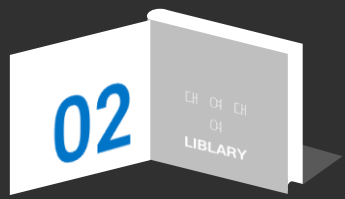
자료명 : 가면 산장 살인사건	장르 : 범죄 추리	재고 : 5	자료위치 : A-7
제한연령 : 12	출판일 : 2014.09.26		
설명 : 강도와 인질 간의 숨 막히는 줄다리기!			
대여횟수 : 0	관심수 : 0		

## 아이템 정보 검색 메서드

- 정상적인 index 값은 0~이상의 값, 찾지 못했을 때 구분 값 -1을 사용
- 배열의 반복으로 title값을 비교해서 index 값을 찾도록 구현

## 아이템 정보 수정 메서드

- 검색 메서드의 title값을 비교해서 index값을 찾도록 구현
- 조건문을 사용하여 index 번호가 0보다 작으면 아이템 정보를 찾지못했다는 메시지를 불러오도록 함
- 수정하고자 하는 자료위치에 새로운 index값을 넣어 초기화함



## 기능별 소개

### 3) 관리자 매니저 \_회원/ 자료/ 대여내역관리

```
//정보 입력단계-----
//책 정보 입력
public Book createBookInfo() throws InputMismatchException {

    Book info = null;
    String title = null;
    String genre = null;
    String localData = null;
    String author = null;
    int limitAge = 0;
    String story = null;
    String launchDate = null;
    // 기본정보 수집: 책이름, 장르, 책위치, 저자, 연령제한, 설명, 출판일

    System.out.println("책이름을 입력해주세요.");
    title = checkVal();

    System.out.println("장르를 입력해주세요.");
    genre = checkVal();

    System.out.println("자료위치를 입력해주세요.");
    localData = checkVal();

    System.out.println("저자를 입력해주세요.");
    author = checkVal();
    while (true) {
        try {
            System.out.println("연령제한 입력해주세요.");
            limitAge = sc.nextInt();
            checkVal();
        } catch (InputMismatchException e) {
            System.out.println("연령제한을 잘못입력했습니다. \n다시 입력해주세요. (숫자입력)");
            checkVal();
            continue;
        }
    }

    System.out.println("설명을 입력해주세요.");
    story = checkVal(); //설명
    System.out.println("출판일을 입력해주세요.");
    launchDate = checkVal(); //출판일

    info = new Book(title, genre, localData,
                    author, limitAge, story, launchDate);

    //배열에 추가
    books.add(info);
    return info;
}
```

```
1. 도서 추가 | 2. 도서수정 | 3. 도서삭제 | 4. 뒤로가기
1
책이름을 입력해주세요.
가면 산장 살인사건
장르를 입력해주세요.
범죄 추리
자료위치를 입력해주세요.
공백을 입력하셨습니다. 다시 입력해주세요.
A-6
저자를 입력해주세요
히가시노 게이고
연령제한 입력해주세요
십이세
연령제한을 잘못입력했습니다.
다시 입력해주세요. (숫자입력)
연령제한 입력해주세요
12
설명을 입력해주세요
강도와 인질 간의 숨 막히는 줄다리기!
출판일을 입력해주세요
2014.09.26
|도서관리 페이지 입니다.
책 전체정보 출력
```

## 정보입력 배열 추가 메서드

-아이템 클래스 타입의 메서드

-클래스 타입의 변수 초기화

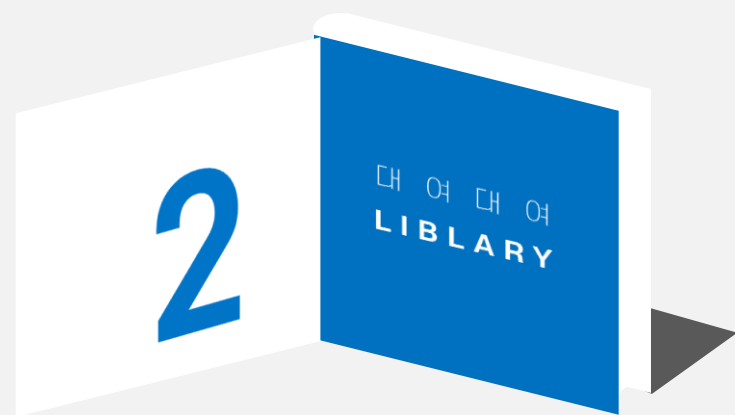
-클래스 내의 변수들을 초기화

-정보 수집단계에서 정보를 입력 받을 때 checkVal(); 이라는 메서드를 사용하여 정보 입력 사항을 입력하지 않고 넘어가지 않도록 함

-연령제한에 인트값을 예외처리 함으로써 숫자입력만 받도록 함

-아이템 정보들을 배열에 추가





## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

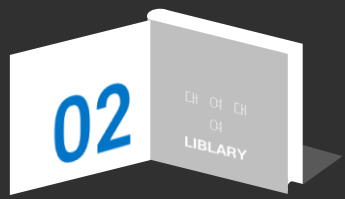
회원, 자료, 대여내역 정보추가

### 3) 관리자매니저

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저 \_박혜미

검색 / 대여 / 예약 / 반납 / 연장 / 연체



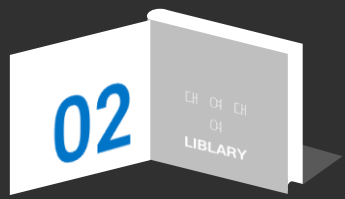
## 기능별 소개

### 4) 대여관리 매니저 \_검색/대여/예약/출력

```
// -----  
public class MemberManager {  
  
    String title = null; // 자료명  
    private ArrayList<RentalList> rentalList;  
  
    // 회원리스트, 자료리스트 불러오기  
    AdminManager adm = AdminManager.getInstance();  
  
    // 기본생성자, 대여리스트 생성  
    public MemberManager() {  
        rentalList = new ArrayList<>();  
  
        // 대여리스트 항목 추가  
        userList uList = new userList();  
        userList2 uList2 = new userList2();  
        userList3 uList3 = new userList3();  
        userList4 uList4 = new userList4();  
        userList5 uList5 = new userList5();  
        userList6 uList6 = new userList6();  
        userList7 uList7 = new userList7();  
        userList8 uList8 = new userList8();  
        userList9 uList9 = new userList9();  
        userList10 uList10 = new userList10();  
        userList11 uList11 = new userList11();  
        userList12 uList12 = new userList12();  
        userList13 uList13 = new userList13();  
        userList14 uList14 = new userList14();  
        userList15 uList15 = new userList15();  
        userList16 uList16 = new userList16();  
  
        this.rentalList.add(uList);  
        this.rentalList.add(uList2);  
        this.rentalList.add(uList3);  
        this.rentalList.add(uList4);  
        this.rentalList.add(uList5);  
        this.rentalList.add(uList6);  
        this.rentalList.add(uList7);  
        this.rentalList.add(uList8);  
        this.rentalList.add(uList9);  
        this.rentalList.add(uList10);  
    }  
}
```

## RentalList 클래스 작성.인스턴스 생성, 리스트 추가

```
public class RentalList {  
  
    public String id;           // 회원ID  
    public String title;        // 자료명  
    public String rentInfo;     // 대여상태  
    public String rentalDate;   // 대여일#  
    public String returnDate;  // 반납예정일#  
    public String returnLimit;  // 반납일#  
    public String reservDate;   // 예약일#  
    public String reservId;     // 예약자  
    public String extendDate;   // 연장된 반납일#  
    public String overdue;      // 연체일  
  
    AdminManager admManager=AdminManager.getInstance();  
  
    RentalList(String id, String title, String start, String end){  
  
        this.id=id;           // 대여한 회원ID  
        this.title=title;      // 대여한 자료명  
        this.rentInfo="대여중"; // 대여상태-----  
        this.rentalDate=start; // 대여일#  
        this.returnLimit=end;   // 반납예정일#  
        this.returnDate=" ";    // 반납일#-----  
        this.reservDate=" ";    // 예약일#  
        this.reservId=" ";      // 예약자  
        this.extendDate=" ";    // 연장된 반납일#  
        this.overdue=" ";       // 연체기간#-----  
  
    }  
  
    public RentalList(){  
  
    }  
  
    public void showRentalListInfo() {  
        System.out.println("대여 ID : "+id+"\t\t대여 자료명 : "+title+"\t\t대여상태 : "+rentInfo);  
        System.out.println("대여일   : "+rentalDate+"\t\t\t반납예정일 : "+returnLimit+"\t\t\t반납일 : "+returnDate);  
        System.out.println("연체기간 : "+overdue+"\t\t\t연장기간 : "+extendDate);  
        System.out.println("예약일   : "+reservDate+"\t\t\t예약자 : "+reservId);  
        System.out.println("-----");  
    }  
}
```



## 기능별 소개

### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// 메인메뉴 1 : 검색/ 대여
// -> 도서/DVD/게임 선택 -> 선택된 자료전체의 기본정보 자동출력 -> 자료명 선택 -> 선택한 자료명 상세정보출력
// 예약기능
void searchItemInfo() {
    int selectNum;

    while (true) {

        System.out.println("1.도서 | 2.DVD | 3. 게임");
        System.out.println("검색하고자 하는 자료의 번호를 입력해주세요.");

        try {

            selectNum = adm.sc.nextInt();
            adm.sc.nextLine();

        } catch (Exception e) { // 생각치 못한 오류발생이 있을 수 있기 때문에.
            System.out.println("잘못된 메뉴입력입니다. \n 확인하시고 다시 입력해주세요");
            adm.sc.nextLine();
            continue;
        }

        switch (selectNum) {
            case 1: // Book 선택시
                System.out.println(
                    "=====
                adm.showBookBasic();
                System.out.println();

                System.out.println("찾으시는 도서명을 입력해주세요.");
                this.title = adm.sc.nextLine();

                int index = adm.searchBookInfo(title);

                if (index < 0) {
                    System.out.println("검색하신 자료의 정보가 없습니다.");
                    continue;
                } else {
                    System.out.println(
                        "=====

                    adm.getBooks().get(index).showAllinfo();

                    while (true) {
                        System.out.println("1.대여 | 2.예약 | 3.뒤로가기 | 4.메인화면 ");
```

## 검색/대여 >> 자료별로 검색, 대여 및 예약 선택

검색은 비회원도 가능. 대여예약시- 회원인증 필요.

로그인 또는 회원가입 후 이용가능.

로그인시 넘어가도록 로그인 체크기능 요청.

자료별로 검색하는 메뉴선택. 메뉴입력 예외처리

자료별로 메뉴선택시. 검색한 자료 없는 경우/있는 경우 구분.

-대여, 예약, 뒤로가기, 메인화면 메뉴선택으로 사용자 편의성.

자료 선택시 자료 기본정보 출력. 상세정보출력

(관리자매니저 자료관리 부분 메서드 호출)

1.도서 | 2.DVD | 3. 게임  
검색하고자 하는 자료의 번호를 입력해주세요.

1

=====

책 기본 정보 출력

자료명 : 더 해빙

장르 : 자기개발

재고 : 4

자료위치 : A-1

저 자 : 이서윤,홍주연

-----

자료명 : 자리의 힘

장르 : 역사

재고 : 4

자료위치 : A-2

저 자 : 팀 마샬

-----

자료명 : 1cm 다이빙

장르 : 시

재고 : 0

자료위치 : A-3

저 자 : 태수,문정

-----

자료명 : 5년후 나에게

장르 : 자기개발

재고 : 4

자료위치 : A-4

저 자 : 포터 스타일

-----

자료명 : 언컨택트

장르 : 경제/경영

재고 : 3

자료위치 : A-5

저 자 : 김용섭

-----

찾으시는 도서명을 입력해주세요.

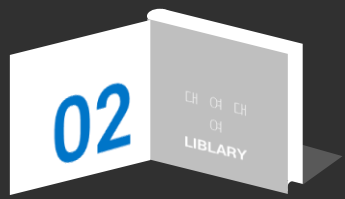
└

검색하신 자료의 정보가 없습니다.

1.도서 | 2.DVD | 3. 게임

검색하고자 하는 자료의 번호를 입력해주세요.

## 예외처리



## 기능별 소개

### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
try {
    selectNum = adm.sc.nextInt();
    adm.sc.nextLine();
} catch (Exception e) {
    System.out.println("잘못된 메뉴입력입니다. \n 확인하시고 다시 입력해주세요");
    adm.sc.nextLine();
    continue;
}

switch (selectNum) {

case 1: // 대여
    // 로그인 상태일시 대여메서드 실행
    if (adm.loginCheck()) {
        creatRentalList();
        break;
    } else {
        // 비로그인 시
        System.out.println();
        System.out.println("이용하시려면 로그인을 해 주세요.");
        adm.login();
        creatRentalList();
        break;
    }

case 2: // 예약
    /* 예약 메서드 */
    break;

case 3: // 뒤로가기
    searchItemInfo();
    break;

case 4: // 메인화면
    return;

default:
    System.out.println("메뉴의 숫자를 입력해주세요.");
    continue;
} // switch : case1(Book) : switch end

break;
} // while end
break;
```

1.도서 | 2.DVD | 3. 게임  
검색하고자 하는 자료의 번호를 입력해주세요.

0  
메뉴의 숫자를 입력해주세요.

1.도서 | 2.DVD | 3. 게임  
검색하고자 하는 자료의 번호를 입력해주세요.

w  
잘못된 메뉴입력입니다.

확인하시고 다시 입력해주세요

1.도서 | 2.DVD | 3. 게임  
검색하고자 하는 자료의 번호를 입력해주세요.

1

책 기본 정보 출력

자료명 : 더 해빙

장르 : 자기개발

재고 : 5

자료위치 : A-1

저 자 : 이서윤,홍주연

자료명 : 자리의 힘

장르 : 역사

재고 : 4

자료위치 : A-2

저 자 : 팀 마샬

자료명 : 1cm 다이빙

장르 : 시

재고 : 0

자료위치 : A-3

저 자 : 태수,문정

자료명 : 5년후 나에게

장르 : 자기개발

재고 : 4

자료위치 : A-4

저 자 : 포터 스타일

자료명 : 언컨택트

장르 : 경제/경영

재고 : 3

자료위치 : A-5

저 자 : 김용섭

찾으시는 도서명을 입력해주세요.

더 해빙

자료명 : 더 해빙

장르 : 자기개발

재고 : 5

자료위치 : A-1

제한연령 : 0

출판일 : 2020.03.01

설명 : 마음가짐을 다듬으며 원하는 것을 얻는 놀라운 가르침!부와 행운을 만나는 출발점, 마법의 감정 HAVING! 국내 최초로 미국에서 선(先)출간되어 세계가 먼저 찾아 읽은 『더 해빙(THE HAVING)』. 세계적인 부자들이 성공 비결로 꼽았다는 ‘운’. 행운은 타고나는 것일까? 소수에게만 주어지는 것일까? 물려받은 재산도, 특출난 재능도 없는 사람도 부자가 될 수 있을..

대여횟수 : 0

관심수 : 0

1.대여 | 2.예약 | 3.뒤로가기 | 4.메인화면

1

hoon님 < 더 해빙 > 자료가 대여완료 되었습니다.

대여일 : 2020-05-12 | 반납예정일 : 2020-05-19

로그인 체크 후,  
로그인 되어있을 시, 대여완료.

1.대여 | 2.예약 | 3.뒤로가기 | 4.메인화면

1

이용하시려면 로그인을 해 주세요.

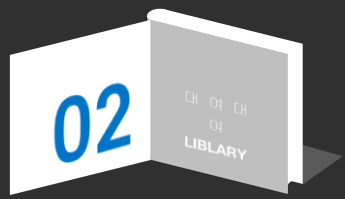
아이디를 입력해 주세요.

mi

비밀번호를 입력해 주세요.

mi

로그인 체크 후,  
로그인 안되어있을 시, 로그인으로



## 기능별 소개

### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// 로그인한 id의 정보를 반환하는 메서드
```

```
Member getloginIdInfo() {
```

```
// 로그인한 회원 Id의 인덱스 찾고,
```

```
int index = adm.loginCheckIndex();
```

```
// 그 회원의 정보 반환
```

```
Member loginIdInfo = null;
```

```
loginIdInfo = adm.getMember().get(index);
```

```
return loginIdInfo;
```

```
}
```

```
// rentalList에 인스턴스 추가
```

```
void addRental(RentalList info) {
```

```
rentalList.add(info);
```

```
}
```

```
// rentalList 인스턴스 생성
```

```
void creatRentalList() {
```

```
// rentalList 객체 생성
```

```
RentalList info = null;
```

```
// 로그인한 id의 회원정보와 id 가져오기
```

```
Member loginIdInfo = getloginIdInfo();
```

```
String id = loginIdInfo.getId();
```

```
// 1. 대여불가여부 확인 : 만약에 회원의 자료 대여개수가 5개일 때
```

```
if (loginIdInfo.numOfRent == 5) {
```

```
System.out.println("자료 대여가능개수(5개)를 모두 사용중입니다. 자료를 반납하신 후 대여를 이용해 주시
```

검색/대여 >> 로그인한 id정보, 검색한 타이틀,  
대여일, 반납일로 객체생성하고 대여내역에 추가

로그인 한 회원이 이용하므로, 회원아이디 반환해주는 메서드 별도구성.

먼저, 대여불가 조건 확인

- 대여권수 5일때, 연체가 있을 때 연체일수만큼 대여불가

게임 기본 정보 출력

자료명 : 젤다의 전설

장르 : 오픈에어 어드벤처

재고 : 5

자료위치 : G-1

제작사 : 닌텐도

자료명 : 슈퍼마리오

장르 : 모험

재고 : 0

자료위치 : G-2

제작사 : 닌텐도

자료명 : 포켓몬스터

장르 : 롤플레이

재고 : 5

자료위치 : G-3

제작사 : 닌텐도

자료명 : 스피드레이2

장르 : 슈팅게임

재고 : 5

자료위치 : G-4

제작사 : 닌텐도

자료명 : 별의커비

장르 : 액션게임

재고 : 5

자료위치 : G-5

제작사 : 닌텐도

찾으시는 Game명을 입력해주세요.

포켓몬스터

자료명 : 포켓몬스터

장르 : 롤플레이

재고 : 5

자료위치 : G-3

제한연령 : 0

출시일 : 2019.11.15

설명 : 《포켓몬스터 소드·실드》는 게임프리크 제작, 포켓몬 컴퍼니 및 닌텐도 배급의 롤플레이 게임이다.

닌텐도 스위치 전용. 포켓몬 연작의 제8세대를 여는 작품이며 더불어 《레츠고! 피카츄·이브이》의 다음가는 프랜차이즈의 두 번째 가정용 콘솔 출시작이다.

대여횟수 : 0

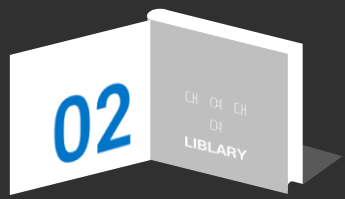
관심수 : 0

1. 대여 | 2. 예약 | 3. 뒤로가기 | 4. 메인화면

1

자료 대여가능개수(5개)를 모두 사용중입니다. 자료를 반납하신 후 대여를 이용해 주세요.

대여가능한 자료수를 모두 대여중일 때,  
대여불가.



## 기능별 소개

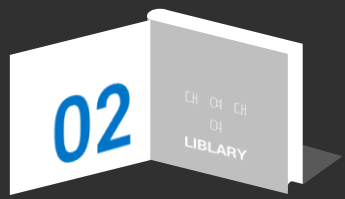
### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// -----  
// 2. 대여가능일때  
} else {  
  
    // 자료 카운트 변경  
    itemRentalCount();  
  
    // 회원 카운트 변경 : 로그인한 아이디로 회원정보 받아서 카운트 변경  
    loginIdInfo.numOfRent += 1; // 회원정보 : 대여자료수 +1  
    loginIdInfo.rentalAvail -= 1; // 회원정보 : 대여가능 자료수 +1  
  
    // 대여일 생성  
    LocalDate rentalDate = LocalDate.now();  
    String start = rentalDate.toString();  
  
    // 반납일 생성 : 대여일+7일  
    LocalDate returnDate = rentalDate.plusDays(7);  
    String end = returnDate.toString();  
  
    // MemberManager 객체 생성  
    info = new RentalList(id, title, start, end);  
  
    // MemberManager 객체 -> 대여리스트에 추가 메서드 1-1. 호출.  
    addRental(info);  
    System.out.println(id + "님 < " + title + " > 자료가 대여완료 되었습니다. ");  
    System.out.println("대여일 : " + start + " | 반납예정일 : " + end);  
    return;  
  
} // else end  
  
} // creatRentalList() end
```

검색/대여 >> 로그인한 id정보, 검색한 타이틀,  
대여일, 반납일로 객체생성하고 대여내역에 추가

대여 가능시, 검색 후 선택한 자료명, 로그인한 아이디,  
대여일(오늘날짜), 반납일(대여일+7)을 입력하여 인스턴스 생성 후  
RentalList에 추가.

회원 카운트변경 - 대여권수+1, 대여가능권수 -1  
자료(자료카운트 메서드로 따로구성) - 대여횟수+1, 재고 -1



## 기능별 소개

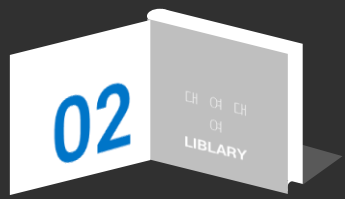
### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// -----  
// 선택한 자료의 타입을 확인하고 카운트변경 (대여횟수, 재고수, 재고수확인후 대여불가처리)  
void itemRentalCount() {  
  
    int index = 0;  
  
    // 입력받은 title이 Book일때!  
    if (adm.searchBookInfo(this.title) >= 0) {  
  
        // 재고가 있으면, 재고수 -1, 대여횟수 +1  
        if (adm.getBooks().get(index).numOfItem > 0) {  
            index = adm.searchBookInfo(this.title);  
            adm.getBooks().get(index).numOfItem -= 1;  
            adm.getBooks().get(index).rentalCount += 1;  
  
            // 대여불가여부 확인 : 자료의 재고가 0일때 -> 자료 예약안내  
        } else if (adm.searchBookInfo(this.title) == 0) {  
            System.out.println("선택하신 자료가 현재 모두 대여중입니다.");  
            System.out.println("(...대여예약 기능을 준비중입니다...)");  
            searchItemInfo();  
        }  
  
        // 입력받은 title이 Dvd일때!  
    } else if (adm.searchDvdInfo(this.title) >= 0) {  
  
        // 재고가 있으면, 재고수 -1, 대여횟수 +1  
        if (adm.getDvd().get(index).numOfItem > 0) {  
            index = adm.searchDvdInfo(this.title);  
            adm.getDvd().get(index).numOfItem -= 1;  
            adm.getDvd().get(index).rentalCount += 1;  
  
            // 대여불가여부 확인 : 자료의 재고가 0일때 -> 자료 예약안내  
        } else if (adm.searchDvdInfo(this.title) == 0) {  
            System.out.println("선택하신 자료가 현재 모두 대여중입니다.");  
            System.out.println("(...대여예약 기능을 준비중입니다...)");  
            searchItemInfo();  
        }  
    }  
}
```

## 검색/대여 >> 자료정보 확인 및 변경

대여시 필요한 자료카운트 변경 부분을 별도 메서드로 작성.  
선택한 타이틀의 자료구분 후(도서, DVD, game)  
재고 없으면, 대여불가, 예약안내(예약기능 미완성..)  
재고 있으면, 대여가능.카운트 (-재고수-1, 대여횟수 +1)





## 기능별 소개

### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// -----  
// 전체 rentalList 에서 title로 검색하기 -> 인덱스 반환  
int searchRentalIndexTitle(String title) {  
  
    int searchRentalIndex = -1;  
  
    for (int i = 0; i < rentalList.size(); i++) {  
        if (rentalList.get(i).title.equals(title)) {  
            searchRentalIndex = i;  
        }  
    }  
    return searchRentalIndex;  
}  
  
// -----  
// 나의 rentalList 에서 title로 검색하기(내 대여내역) -> 인덱스 반환  
int checkTitle(String title) {  
  
    String id = getloginIdInfo().getId();  
  
    int index2 = -1;  
    for (int i = 0; i < rentalList.size(); i++) {  
        if (rentalList.get(i).id.equals(id) && rentalList.get(i).title.equals(title)) {  
            index2 = i;  
            break;  
        }  
    }  
    return index2;  
}  
  
// -----  
// rentalList 에서 id로 검색하기(내 대여내역) -> 인덱스 반환  
void showMyRentalList() {  
    Member loginIdInfo = getloginIdInfo();  
    String id = getloginIdInfo().getId();  
  
    System.out.println("나의 대여 내역");  
    System.out.println("대여중인 자료수 : " + loginIdInfo.numOfRent + " | 대여가능 자료수 : " + 1  
        + " | 연장가능 횟수 : " + loginIdInfo.numOfExtens + " | 연체일수 : " + loginIdInfo.  
        "=====");  
  
    for (int i = 0; i < rentalList.size(); i++) {  
        if (rentalList.get(i).id.equals(id)) {  
  
            rentalList.get(i).showRentalListInfo();  
        }  
    }  
}
```

## 다양한 출력 및 반환 메서드 : RentalList 회원용

나의 대여내역 확인.

더하여 나의 대여내역 중 자료명으로 검색 후 인덱스 반환.

대여내역 중 검색한 자료명 인덱스 반환.

나의 대여 내역

대여중인 자료수 : 4 | 대여가능 자료수 : 1 | 연장가능 횟수 : 1 | 연체일수 : 0

나의 대여내역정보

대여 ID : mi      대여 자료명 : 슈퍼마리오      대여상태 : 대여중  
대여일 : 2020-05-07      반납예정일 : 2020-05-12      반납일 :  
연체기간 :      연장기간 :  
예약일 : 2020-05-13      예약자 : hoon

대여 ID : mi      대여 자료명 : 언컨택트      대여상태 : 대여중  
대여일 : 2020-05-07      반납예정일 : 2020-05-14      반납일 :  
연체기간 :      연장기간 :  
예약일 :      예약자 :

대여 ID : mi      대여 자료명 : 자리의 힘      대여상태 : 반납완료  
대여일 : 2020-05-07      반납예정일 : 2020-05-14      반납일 : 2020-05-12  
연체기간 :      연장기간 :  
예약일 :      예약자 :

대여 ID : mi      대여 자료명 : 1cm 다이빙      대여상태 : 대여중  
대여일 : 2020-05-07      반납예정일 : 2020-05-14      반납일 :  
연체기간 :      연장기간 :  
예약일 : 2020-05-14      예약자 : hoon

대여 ID : mi      대여 자료명 : 5년 후 나에게      대여상태 : 대여중  
대여일 : 2020-05-07      반납예정일 : 2020-05-14      반납일 :  
연체기간 :      연장기간 :  
예약일 :      예약자 :

반납이나 연장하실 자료명을 입력해주세요.

슈퍼마리오

1. 반납 | 2. 연장 | 3. 메인화면

2

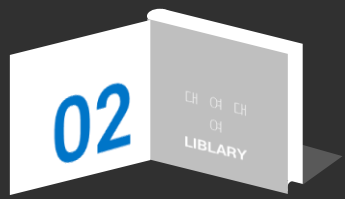
hoon

예약자가 있어 연장할 수 없습니다.

1. 반납 | 2. 연장 | 3. 메인화면

예약자가 있을시 연장불가





## 기능별 소개

### 4) 대여관리 매니저 \_ 검색/대여/예약/출력

```
// -----  
public void showAllRentalListInfo() {  
    for (int i = 0; i < rentalList.size(); i++) {  
        rentalList.get(i).showRentalListInfo();  
        System.out.println(  
            "-----"  
        );  
    }  
}  
  
// -----  
// rentalList 에서 id로 검색하기(관리자사용) -> 대여내역 출력  
void showIDRentalList() {  
  
    System.out.println("찾아보실 회원id를 입력해주세요.");  
    String id = adm.sc.nextLine();  
  
    for (int i = 0; i < rentalList.size(); i++) {  
        if (rentalList.get(i).id.equals(id)) {  
  
            rentalList.get(i).showRentalListInfo();  
  
        }  
    }  
}  
  
// -----  
// rentalList 에서 title로 검색하기(관리자사용) -> 인덱스 반환  
void showTitleRentalList() {  
  
    System.out.println("찾아보실 자료명을 입력해주세요.");  
    String title = adm.sc.nextLine();  
  
    for (int i = 0; i < rentalList.size(); i++) {  
        if (rentalList.get(i).title.equals(title)) {  
  
            rentalList.get(i).showRentalListInfo();  
  
        }  
    }  
}
```

## 다양한 출력메서드 : RentalList 관리자용

대여내역관리 페이지에서 전체보기, 회원id로 검색해서 보기, 자료명으로 검색해서 보기,를 활용해 대여내역관리 가능.

대여내역관리 페이지 입니다.

1. 전체 대여내역보기 | 2. 회원 id로 대여내역보기 | 3. 자료명으로 대여내역보기 | 4. 뒤로가기

2

찾아보실 회원id를 입력해주세요.

min

### 회원 id 검색으로 대여내역출력

대여 ID : min	대여 자료명 : 슈퍼마리오	대여상태 : 대여중
대여일 : 2020-05-06	반납예정일 : 2020-05-13	반납일 :
연체기간 :	연장기간 :	
예약일 :	예약자 :	

대여 ID : min	대여 자료명 : 1cm 다이빙	대여상태 : 대여중
대여일 : 2020-05-01	반납예정일 : 2020-05-08	반납일 :
연체기간 :	연장기간 : 2020-05-15	
예약일 :	예약자 :	

1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃

대여내역관리 페이지 입니다.

1. 전체 대여내역보기 | 2. 회원 id로 대여내역보기 | 3. 자료명으로 대여내역보기 | 4. 뒤로가기

3

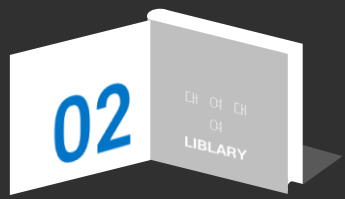
찾아보실 자료명을 입력해주세요.

자리의 힘

### 자료명 검색으로 대여내역출력

대여 ID : mi	대여 자료명 : 자리의 힘	대여상태 : 대여중
대여일 : 2020-05-07	반납예정일 : 2020-05-14	반납일 :
연체기간 :	연장기간 :	
예약일 :	예약자 :	

1. 회원관리 | 2. 도서관리 | 3. DVD관리 | 4. 게임관리 | 5. 대여내역관리 | 6. 로그아웃



## 기능별 소개

### 4) 대여관리 매니저 \_검색/대여/예약/출력

#### 리팩토링 관련

\_작업중 리스트부분을 별도클래스로 분리하여 가져오면서

—> 대여관리매니저를 재조정

\_메인라인에서 대여관리 관련하여 분기되는 흐름을

—> 대여관리매니저로 가져와 메인간소화

\_자료타입(책, DVD, 게임)에 대한 리스트구조 다형성 구현 필요성

—> 문법속지 및 시간 부족으로 적용실패 —> 추후 보완가능예상

\_대여리스트에서 반납완료되는 리스트는 별도로 이동하여 저장.

대여리스트의 최적화 유지. —> 추후 보완가능 예상

#### 세부기능 및 내용관련

\_기준이 되는 대여일 생성에서 현재 날짜에 대한 문제

—> Date, Calender 문법의 숙지어려움..—> java.time에 대한

문법공부, 적용. —> 현재날짜 생성 후 고정안되는 어려움.

—> toString 메서드로 해결. 카운트에 대한 처리 완료

—> 예약기능 미완성.. 문법숙지 후 보완가능 예상

\_대여, 반납 기능을 구현하며 회원변수, 자료변수, 대여관리리스트 변수에 대한 관계 다시 조정.

\_접근제어자의 중요성..

## 기능 구현시 고민.. 문제..해결..

#### 프로그램 흐름관련 \_회원관점

\_어떤 방법으로 흐름을 정리하며 기능을 실행해야

회원이 사용하기에 편할지 고민.

Ex. 기본정보/상세정보, 내 대여내역, 우선시 해야하는 정보 선별.

뒤로가기. 메인으로가기 등

#### 데이터 활용관련 \_관리자관점

\_관리자가 대여리스트를 통해 어떤 선별정보들을 얻고,

어떤 관리기능들을 할 수 있을지 고민.

Ex. 대여횟수로 인기자료, 추천자료 안내하고, 재고 늘릴 수 있을 것.

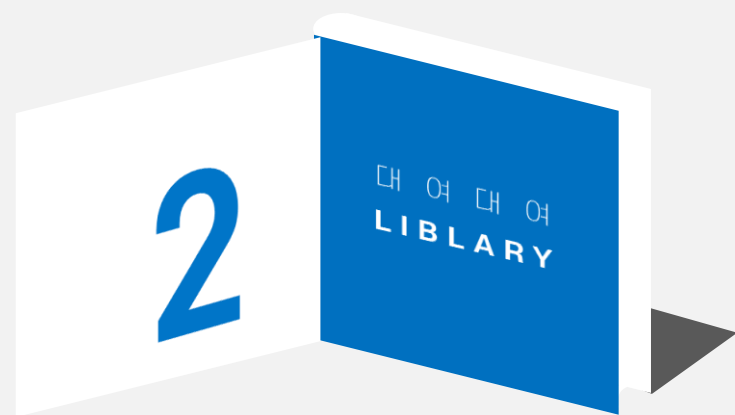
예약이 많은 자료 찾고, 재고 늘릴 수 있을 것.

연체를 많이 하는 회원 분류하여 제재가능.

대여를 많이 하는 우수회원 레벨생성 가능.

반납일 하루전인 자료들 모아서 회원에게 반납안내문자 보낼 수 있을 것.

등등...



## ◆ 기능별 소개

### 1) 메인

### 2) 회원정보/자료정보

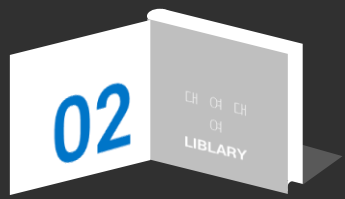
회원, 자료, 대여내역 정보추가

### 3) 관리자매니저

회원관리 / 자료관리 / 대여내역관리

### 4) 대여관리매니저 \_임재영

검색 / 대여 / 예약 / 반납 / 연장 / 연체



## 기능별 소개

### 4) 대여관리 매니저 \_반납/ 연장/ 연체

```
// 반납
void itemReturn(int index) {

    // 만약에 대여내역에 같은 자료명이 한개 이상일 때,
    //
    // 반납일이 null이고,

    // 반납일 = 현재날짜로 생성
    LocalDate today = LocalDate.now();
    String returnDate = today.toString();

    // 나의 대여리스트 중 해당 인덱스의 반납일이 변경됨.
    rentalList.get(index).returnDate = returnDate;
    rentalList.get(index).rentInfo = "반납완료";

    // 나의 대여리스트 중 해당 인덱스 출력. 확인.
    rentalList.get(index).showRentalListInfo();

    // 회원 카운트 변경 : 로그인한 아이디로 회원정보 받아서 카운트 변경
    Member loginIdInfo = getloginIdInfo();
    loginIdInfo.numOfRent -= 1; // 회원정보 : 대여한 자료수 -1
    loginIdInfo.rentalAvail += 1; // 회원정보 : 대여가능 자료수 +1
    // 자료 카운트 변경 : 반납했으니 다시 재고 +1
    int itemIndex = 0;

    // 자료가 책이면,
    if (adm.searchBookInfo(this.title) >= 0) {
        itemIndex = adm.searchBookInfo(this.title);
        adm.getBooks().get(itemIndex).numOfItem += 1;

        // 자료가 DVD면,
    } else if (adm.searchDvdInfo(this.title) >= 0) {
        itemIndex = adm.searchDvdInfo(this.title);
        adm.getDvd().get(itemIndex).numOfItem += 1;

        // 자료가 Game이면,
    } else if (adm.searchGameInfo(this.title) >= 0) {
        itemIndex = adm.searchGameInfo(this.title);
        adm.getGame().get(itemIndex).numOfItem -= 1;
    }

    System.out.println(gap+"일");

    if( gap > 7) { // 연체.

        adm.getMember().get(adm.loginCheckIndex()).rentalAvail = 0; // 대여가능권수
        adm.getMember().get(adm.loginCheckIndex()).numOfExtens = 0; // 연장가능횟수
        //numOfRent = ?? // 대여권수, 빌려간게 더 있을수도 있고 없을수도있고. ??

        System.out.println(title + "연체일수가 있습니다.");
        System.out.println(title + "연체일수는 " + gap + " 일입니다.");
        System.out.println(overdue + "일 동안 자료를 대여하실 수 없습니다.");
        System.out.println("처리 : " + today);
    }else if(gap < 7) { // 연체 없음.

        returnDate = today;

        System.out.println(title+"자료가 정상적으로 반납되었습니다.");
        System.out.println("처리 : " + today);

        adm.getMember().get(adm.loginCheckIndex()).numOfRent --; // 대여권수
        adm.getMember().get(adm.loginCheckIndex()).rentalAvail ++; // 대여가능권수
        adm.getMember().get(adm.loginCheckIndex()).numOfExtens = 1; // 연장가능횟수

    }

    // 대여가능으로 바꾼다.
    rentalList.get(index).title
```

반납 >> 내 대여내역에서, 타이틀 검색 후 반납.

대 여 대 여 LIBRARY

1. 검색/대여 | 2.로그인 | 3.이용안내/회원가입 | 4.내 대여내역(반납/연장) | 5.로그아웃 | 6.관리자페이지 | 7.프로그램 종료

#### 4

나의 대여 내역

대여중인 자료수 : 1 | 대여가능 자료수 : 4 | 연장가능 횟수 : 1 | 연체일수 : 0

대여 ID : wow	대여 자료명 : 더 해빙	대여상태 : 대여중
대여일 : 2020-05-12	반납예정일 : 2020-05-19	반납일 :
연체기간:	연장기간 :	
예약일 :	예약자 :	

반납이나 연장하실 자료명을 입력해주세요.

더 해빙

1. 반납 | 2.연장 | 3.메인화면

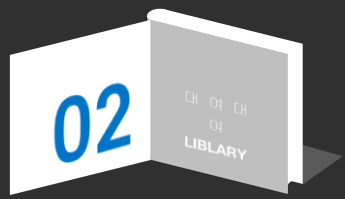
1

대여 ID : wow	대여 자료명 : 더 해빙	대여상태 : 반납완료
-------------	---------------	-------------

대여일 : 2020-05-12	반납예정일 : 2020-05-19	반납일 : 2020-05-12
------------------	--------------------	------------------

연체기간:	연장기간 :
-------	--------

예약일 :	예약자 :
-------	-------



## 기능별 소개

### 4) 대여관리 매니저 \_반납/ 연장/ 연체

```
// 연장.
void extention(int index) {

    Member loginIdInfo = getloginIdInfo();
    System.out.println(rentallList.get(index).reservId);

    // 만약에 내 연장횟수가 0이면, 연장불가
    if (loginIdInfo.numOfExtens == 0) {
        System.out.println("연장가능 횟수가 0입니다.");

        // 만약에 연장하려는 내 대여내역에 예약자가 있으면, 연장불가.
    } else if (rentallList.get(index).reservId != " ") {
        System.out.println("예약자가 있어 연장할 수 없습니다. ");

        // 연체가 있으면, 연장불가
    } else if (loginIdInfo.overdue > 0) {
        System.out.println("대여하신 자료 중 연체가 있어 연장을 할 수 없습니다.");

        // 연장이 되면 -> 연장된 반납일 생성, 카운트 변경
    } else {
        // 연장된 반납일 생성 : 나의 대여리스트 중 해당 인덱스의 반납예정일 + 7
        LocalDate returnLimit0 = LocalDate.parse(rentallList.get(index).returnLimit);

        // 나의 대여리스트 중 해당 인덱스의 연장된 반납일 -> 연장된 반납일로 수정
        rentallList.get(index).extendDate = extendDate;
        rentallList.get(index).rentInfo = "연장완료";

        // 회원 카운트 변경 : 로그인한 아이디로 회원정보 받아서 카운트 변경
        loginIdInfo.numOfExtens -= 1; // 회원정보 : 연장가능 횟수 -1

        // 나의 대여리스트 중 해당 인덱스 출력, 확인.
        System.out.println(
            "-----"
        );
        rentallList.get(index).showRentallListInfo();
    }

    long gap = startDate1.getTime() - endDate1.getTime() / (24*60*60*1000);
    gap = Math.abs(gap);

    if((int)gap > 7) { //연체 있을때.
        System.out.println(title + "연체일수가 " + gap + " 일 입니다.");
        System.out.println(gap + " 기간동안 연장하실 수 없습니다.");
        System.out.println("처리 : " + today);

        adm.getMember().get(adm.loginCheckIndex()).rentalAvail = 0; // 대여가능권수
        adm.getMember().get(adm.loginCheckIndex()).numOfExtens = 0; // 연장가능횟수

    }else if((int)gap < 7) { // 연체 없을때

        returnDate = format.format(cal.getTime());
        cal.add(Calendar.DATE, 7);
        String extenDate = format.format(cal.getTime());

        System.out.println(title + "자료가 정상적으로 7일 연장되었습니다.");
        System.out.println(returnDate + " 까지 반납하세요.");
        System.out.println("처리 : " + today);

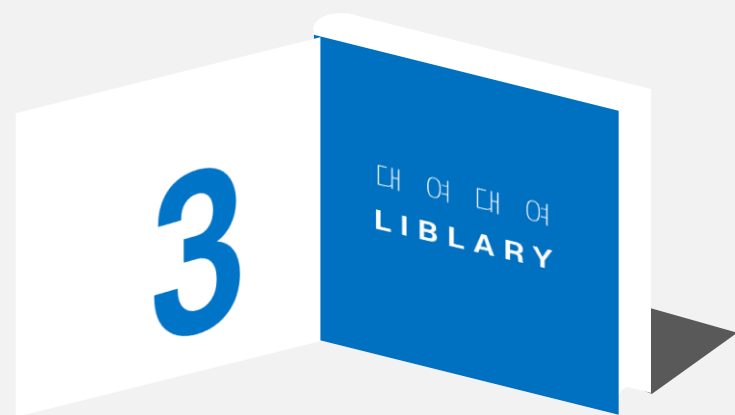
        //admManager.getMember().get(admManager.loginCheckIndex()).numOfRent ??? // 대여권수, 빌렸던 책
        //admManager.getMember().get(admManager.loginCheckIndex()).rentalAvail ??? // 대여가능권수, ???
        adm.getMember().get(adm.loginCheckIndex()).numOfExtens = 0; // 연장가능횟수, 연장을 했으니깐 0으로 바꿔준다.
    }
}
```

연장 >> 내 대여내역에서, 타이틀 검색 후 연장.

1. 반납 | 2.연장 3.메인화면  
2

대여 ID : wow	대여 자료명 : 더 해빙	대여상태 : 연장완료
대여일 : 2020-05-12	반납예정일 : 2020-05-19	반납일 : 2020-05-12
연체기간:	연장기간 : 2020-05-26	
예약일 :	예약자 :	

반납예정일 기준으로 7일 연장.



## ◆ 협업과정에서의 문제점 & 해결방법

### 01\_프로그램 흐름관련

\_로그인 없이 이용가능한 검색과 로그인 후 이용가능한 대여부분에서 계속 로그인 요청을 만나는 문제  
>>> 로그인상태값으로 로그인 체크하고, 조건문을 통해 해결

\_로그아웃 기능 >>> 메인메뉴 줄 하나로 구성하여 단순화함  
\_예외처리, 뒤로가기, 메인화면으로 가기 기능 등으로  
프로그램 흐름에서 사용자 편의성 보완

### 02\_협업관련

\_같은 클래스를 2명이 공동작업할 때, 시스템 오류를 피하기 위한 작업의 불편함 >>> 이클립스 프로젝트의 패키지를 나눠서 작업 후, 마스터 패키지에서 취합하는 과정으로 작업...했으나 그래도 어려움이 있었음. 반복되는 충돌과 재설치 과정..

\_초기계획에서 생각하지 못한 기능, 변수가 생겨서 작업중 변경사항이 많아짐, 작업하면서 다시 팀원들과 상의하고 변경하여 적용하는 작업이 많아짐.

\_명확한 역할분담으로 전체 흐름에 대한 팀원들의 이해도 상이



## ◆ 협업과정에서의 문제점 & 해결방법

\_프로그램 테스트작업 : 수정보완

### 수정완료 수정위치

### 수정내용

#### item 부분

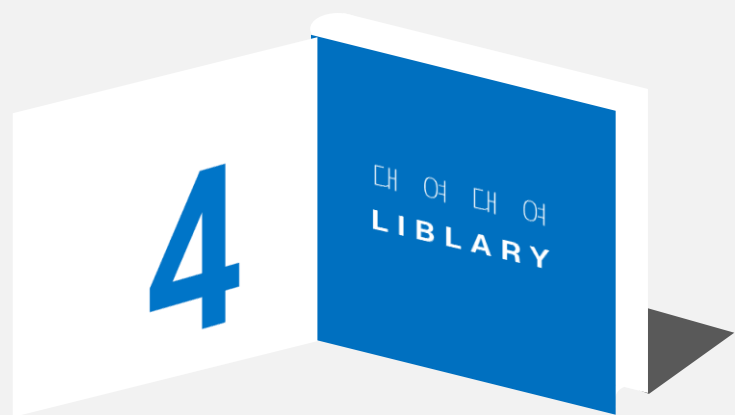
수정완료 게임	1. game에서 포켓몬스터 이름같은.
수정완료 도서관리	2. A-4, A-5 자리바뀜.
수정완료 회원상세정보	3. id가 안보여진다. 보여줘야 수정이 가능하다.

#### adminmanager 부분

수정완료 회원정보수정전체	1. 회원정보수정할때 => 내용없이 엔터치면 그냥 등록됨.
수정완료 관리자페이지	2. 비밀번호3번 틀리면 메인페이지 나올때 다른거 나온다.
수정완료 관리자페이지	3. 회원정보수정 => 수정하고난 다음에 5번 뒤로가기 안먹힌다.
수정완료 회원가입	4. 아이디 한글입력된다 => 영문만 입력되도록 해야된다.
수정완료 회원가입	5. 엔터치면 입력 안해도 다음걸로 넘어간다.
수정완료 검색/대여	6. 타이틀검색하고 => 정보뜬다음에 분기 나왔으면 좋겠다.
수정완료 dvd 추가 , game 추가	7. 책이름으로 뜬다.
수정완료 dvd, game, book 추가	8. 추가 할때, 입력안하고 엔터치면 그냥넘어간다.

#### memberManager 부분

수정완료 내 대여내역	1. 출력에러
수정완료 내 대여내역	2. 대여한거 없을때, 정보뜬 다음에, 뒤로가기 버튼 있었으면 좋겠다. 대여내역이 없는데 반납,연장 물어본다.
추후보완 예약	3. 예약 안됨.
추후보완 대여	4. 재고 없을 때, 대여가 안되도록 수정필요.



## ◆ 프로젝트 개선방향

### 01\_RentalList 활용한 추가기능

- \_자료의 대여횟수, 좋아요수 활용한 인기자료/ 추천자료
- \_반납일 기준으로 리스트 정렬 후 반납일 1일전 안내문자발송
- \_연체자 기준으로 리스트 정렬 후 패널티 적용

### 02\_검색/대여/예약/반납/연장/연체

- \_구조 보완(리스트 분리, 자료유형 구분하여 처리하는 시스템 보완)
- \_자료 인덱스 통한 동일자료 구분
- \_동일자료 대여반납시 중복처리기능 보완
- \_여러 자료 선택해서 대여 또는 반납
- \_예약/연체 기능 보완
- \_제한연령 자료대여에 적용

### 03\_관리자 추가기능

- \_회원이 자료 신청
- \_유료 대여시스템
- \_회원등급에 따른 대여관리시스템 혜택 기능





감사합니다